



UniCEUB – Centro Universitário de Brasília

FAET – Faculdade de Ciências Exatas e Tecnologia

Curso de Engenharia da Computação

Projeto Final

CRISTIANO RABELO ISAAC

COMUNICAÇÃO ENTRE UM VEÍCULO E O USUÁRIO ATRAVÉS DE UM SISTEMA GSM

**BRASÍLIA, DF
2º SEMESTRE DE 2009**

CRISTIANO RABELO ISAAC

**COMUNICAÇÃO ENTRE UM VEÍCULO E O
USUÁRIO ATRAVÉS DE UM SISTEMA GSM**

Monografia apresentada ao Curso de Engenharia da Computação, como requisito parcial para obtenção do grau de Engenheiro de Computação.

Orientador: Prof. José Julimá Bezerra Junior

BRASÍLIA, DF

2º SEMESTRE DE 2009

Agradecimentos

Agradeço a toda a minha família, principalmente aos meus pais, Alexandre Cotta Isaac e Maria Anízia Rabelo Isaac, por todo o apoio, carinho, amor e respeito.

Agradeço ao meu tio Carlinhos, que me ajudou no desenvolvimento do projeto e programação, me dando dicas fundamentais e me incentivou bastante para eu realizá-lo dentro do prazo.

Agradeço à minha namorada, Marina Morato, pela paciência, grande apoio, amor e compreensão durante essa difícil fase.

Agradeço também ao corpo docente do UniCEUB, que contribuiu, através de seus conhecimentos, para o meu aprendizado e crescimento intelectual, possibilitando a realização desse projeto.

Agradeço aos meus amigos, que me ajudaram pessoalmente, psicologicamente e profissionalmente durante essa longa jornada.

Resumo

Este trabalho apresenta o projeto de um sistema de alarme veicular com a vantagem de o condutor ser alertado sobre o disparo do mesmo através do celular. Assim, quando houver uma tentativa de furto ou violação do veículo, algum dos sensores utilizados pelo projeto é acionado, enviando o alerta ao usuário, que pode tomar suas devidas precauções. Para comprovar sua eficiência, foram realizados testes simulando sua utilização em um veículo, utilizando os recursos necessários do mesmo. O projeto foi dividido em módulos para simplificar seu funcionamento, podendo ser utilizado em conjunto com outros sensores e alarmes, facilitando ainda mais sua possível utilização comercial.

Palavras-Chave: Alarme, veículo, microcontrolador, celular, módulo GSM, comandos AT, sensores.

Abstract

This work presents the project of a vehicular alarm system with the advantage that the owner is alerted about the trigger of it through the phone. This way, when an attempted theft or breach of the car occurs, a sensor used by the project will trigger it, sending a warning to the user, which can take his precautions even from far away. To prove its efficiency had been carried out tests simulating his use in a car with the necessary resources. The project will be divided into modules to simplify his operation and can be used in conjunction with other sensors and alarm systems, making easier their possible commercial use.

Keywords: Alarm, vehicle, microcontroller, mobile phone, GSM module, AT commands, sensors.

Sumário

1	INTRODUÇÃO.....	1
1.1	OBJETIVOS.....	1
1.2	VISÃO GERAL.....	3
1.3	ORGANIZAÇÃO DA MONOGRAFIA.....	5
2	REFERENCIAL TECNOLÓGICO.....	7
2.1	ALARME.....	7
2.1.1	<i>Alarmes Veiculares.....</i>	8
2.2	SENSORES VEICULARES.....	10
2.3	PRESTADORAS DE SERVIÇOS VEICULARES.....	15
2.4	MICROCONTROLADOR.....	15
2.4.1	<i>Circuito do kit de estudo.....</i>	19
2.4.2	<i>Portas de Entrada e Saída.....</i>	21
2.4.3	<i>Níveis de tensão.....</i>	23
2.4.4	<i>Interface Serial.....</i>	24
2.5	TERMINAL E MÓDULO WIRELESS GSM.....	25
2.5.1	<i>Níveis de tensão.....</i>	28
2.6	COMANDOS AT.....	29
2.7	LINGUAGEM DE PROGRAMAÇÃO.....	29
3	IMPLEMENTAÇÃO DO PROJETO.....	31
3.1	ADAPTAÇÃO EM PROTOBOARD.....	31
3.2	FONTE.....	31
3.2.1	<i>Bateria recarregável.....</i>	35
3.3	MICROCONTROLADOR.....	37

3.3.1	<i>Oscilador</i>	38
3.3.2	<i>LEDs</i>	39
3.3.3	<i>Sensores simulados</i>	40
3.3.4	<i>Reset</i>	41
3.3.5	<i>Timer e Interrupção</i>	41
3.4	MÓDULO WIRELESS GSM	42
3.5	COMUNICAÇÃO ENTRE O MICROCONTROLADOR E O MÓDULO GSM	44
3.6	SOFTWARE	45
4	RESULTADOS OBTIDOS	52
5	CONCLUSÕES	58
5.1	DIFICULDADES ENCONTRADAS	59
5.1.1	<i>Celular</i>	59
5.1.2	<i>MC35i</i>	61
5.1.3	<i>Níveis de tensão entre o microcontrolador e módulo GSM</i>	63
5.1.4	<i>Comunicação com o Módulo GSM</i>	64
5.2	PROPOSTAS FUTURAS	64
5.2.1	<i>Interação com componentes do veículo</i>	64
5.2.2	<i>Integração com software no Celular (Java)</i>	65
5.2.3	<i>Interação com módulo GPS</i>	65
5.2.4	<i>Interação com sensores específicos</i>	66
	REFERÊNCIAS BIBLIOGRÁFICAS	67
	APÊNDICES	69
	APÊNDICE I – CÓDIGO FONTE DO PROGRAMA	70
	APÊNDICE II – PREÇOS DOS COMPONENTES UTILIZADOS	90
	ANEXOS	92

ANEXO I – CÓDIGO FONTE PARA COMUNICAÇÃO ENTRE O MODULO GSM E MICROCONTROLADOR.....	93
ANEXO II – DATASHEET AT89S52	103
ANEXO III – DATASHEET BENQ M22 REV. 0.2	104
ANEXO IV – BENQ M22 AT COMMAND REV. 0.3.0.....	105
ANEXO V – DATASHEET LM317.....	106
ANEXO VI – DATASHEET MAX232	107

Lista de Figuras

FIGURA 1 – INTERAÇÃO ENTRE O CIRCUITO E DEMAIS DISPOSITIVOS.....	4
FIGURA 2 – SENSOR DE PORTA VEICULAR	10
FIGURA 3 – FUNCIONAMENTO DO SENSOR ULTRA-SOM COM RECEPTOR E TRANSMISSOR.	11
FIGURA 4 – SENSOR DE CHOQUE COM ACELERÔMETRO	12
FIGURA 5 – VISTA GERAL DE UM ALTO-FALANTE E SEUS COMPONENTES	13
FIGURA 6 – CHAVE <i>PUSHBUTTON</i> SEM TRAVA.	15
FIGURA 7 – MICROCONTROLADOR ATMEL AT89S52.....	19
FIGURA 8 – MICROCONTROLADOR AT89S52	22
FIGURA 9 – CIRCUITO INTEGRADO COM MODEM GSM/GPRS BENQ M22.....	27
FIGURA 10 – CIRCUITO DA FONTE QUE RECARREGA A BATERIA E ALIMENTA O CIRCUITO.....	32
FIGURA 11 – CIRCUITO DA FONTE PARA ALIMENTAÇÃO DO MÓDULO GSM	34
FIGURA 12 – CIRCUITO DA FONTE PARA ALIMENTAÇÃO DO MICROCONTROLADOR.....	35
FIGURA 13 – CIRCUITO DO MICROCONTROLADOR.....	38
FIGURA 14 – OSCILADOR DE QUARTZO DE DOIS TERMINAIS.....	39
FIGURA 15 – CIRCUITO DO MÓDULO WIRELESS GSM	43
FIGURA 16 – CIRCUITO PARA COMUNICAÇÃO ENTRE O MICROCONTROLADOR E O MÓDULO .	45
FIGURA 17 – FLUXO OPERACIONAL DE FUNCIONAMENTO	46
FIGURA 18 – CÓDIGO DE ACIONAMENTO PARA VERIFICAÇÃO DOS SENSORES	47
FIGURA 19 – CÓDIGO DA FUNÇÃO DE ATRASO E DA VERIFICAÇÃO DOS SENSORES	49
FIGURA 20 – IMPOSSÍVEL ENVIAR MENSAGEM	51
FIGURA 21 – TENSÃO DA FONTE DE 12VDC	52
FIGURA 22 – TENSÃO DE SAÍDA MEDIDA NO LM350	53
FIGURA 23 – TENSÃO DE SAÍDA NO LM7805	54

FIGURA 24 – TENSÃO DE SAÍDA DO LM317T	54
FIGURA 25 – TENSÃO INICIAL DA BATERIA RECARREGÁVEL	55
FIGURA 26 – TENSÃO DA BATERIA RECARREGADA POR 8 HORAS APROXIMADAMENTE.....	56
FIGURA 27 – CELULAR SIEMENS MODELO S40 E ACESSÓRIOS.	60
FIGURA 28 – CONECTOR PARA ADAPTAÇÃO DO MÓDULO.	62

Lista de Tabelas

TABELA 1 – PINOS DE GRAVAÇÃO PELA PORTA PARALELA PARA O MICROCONTROLADOR. ...	20
TABELA 2 – NÍVEIS DE TENSÃO DE ENTRADA E SAÍDA	24
TABELA 3 – NÍVEIS DE TENSÃO DOS PINOS UTILIZADOS DO MÓDULO GSM.....	28
TABELA 4 – ETAPAS REALIZADAS QUANDO DETERMINADO <i>LED</i> ACENDE	40
TABELA 5 – REGISTRADOR DE INTERRUPÇÕES	42
TABELA 6 – COMPONENTES UTILIZADOS, REFERÊNCIA E VALORES. (CONTINUAÇÃO).	90

Lista de Siglas e Abreviaturas

BIT: Acrônimo do inglês – ***Binary Digit***, Dígito Binário.

bps: Acrônimo do inglês – ***bits per second***, bits por segundo.

CI: ***Circuito Integrado***.

CPU: Acrônimo do Inglês – ***Central Processing Unit***, Unidade Central de Processamento.

DTE: Acrônimo do inglês – ***Data Terminal Equipment***, Equipamento Terminal de Dados.

DCE: Acrônimo do inglês – ***Data Circuit-Terminating Equipment***, Equipamento de Comunicação de Dados.

GPRS: Acrônimo do inglês – ***General Packet Radio Service***, Serviço Geral de Pacotes por Rádio.

GPS: Acrônimo do inglês – ***Global Positioning System***, Sistema de Posicionamento Global.

GSM: Acrônimo do inglês – ***Global System for Mobile Communications***, Sistema Global para Comunicação Móvel.

ISP: Acrônimo do inglês – ***In-System Programming***, Programação dentro do sistema.

LED: Acrônimo do inglês – ***Light Emitting Diode***, Diodo Emissor de Luz.

LPT: Acrônimo do inglês – ***Line Print Terminal***, Terminal de Linha de Impressão, também chamado de Porta Paralela.

M2M: Acrônimo do inglês – ***Machine-to-Machine***, Máquina para máquina.

NMEA: Acrônimo do inglês – ***National Marine Electronics Association***, Associação Nacional Eletrônica da Marinha.

RAM: Acrônimo do inglês – ***Random Memory Access***, Memória de Acesso Aleatório.

ROM: Acrônimo do inglês – ***Read Only Memory***, Memória apenas de Leitura.

SIM: Acrônimo do inglês – ***Subscriber Identity Module***, Módulo de Identificação do Assinante.

SMS: Acrônimo do inglês – ***Short Message Service***, Serviço de Mensagens Curtas.

SMT: Acrônimo do inglês – ***Surfaced Mount Technology***, Tecnologia de montagem em superfície.

USART: Acrônimo do inglês – ***Universal Synchronous Asynchronous Receiver Transmitter***, Interface Universal de Transmissão e Recepção Síncrona e Assíncrona.

USB: Acrônimo do inglês – ***Universal Serial Bus***, Barramento Serial Universal.

ZIF: Acrônimo do inglês – ***Zero Insertion Force***, Força de Inserção Zero.

Lista de Símbolos

<i>B</i>	Byte
<i>V</i>	Tensão elétrica em volts
<i>K</i>	Representação de 10^3
<i>M</i>	Representação de 10^6
<i>m</i>	Representação de 10^{-3}
<i>u</i>	Representação de 10^{-6}
<i>Hz</i>	Frequência em Hertz

1 Introdução

Neste capítulo é descrita a introdução do projeto, apresentando resumidamente seus objetivos, como esta monografia está estruturada, fornecendo uma visão geral do mesmo para facilitar seu entendimento e esclarecer seu foco.

1.1 *Objetivos*

A área de segurança automotiva tem evoluído rapidamente de acordo com a necessidade do usuário se prevenir de eventuais roubos ou furtos, reduzindo a probabilidade de ocorrer um sinistro.

A principal função dos alarmes automotivos é a prevenção de um eventual furto, podendo ser do próprio veículo ou dos bens internos, pois o usuário não estaria presente em seu interior.

A prevenção de roubos ocorre principalmente pela conscientização e cautela do usuário, evitando estacionar ou passar por determinados locais, ter atenção ao entrar ou sair do veículo, dentre outras medidas. Porém, em algumas situações o assalto torna-se inevitável. Nesse caso, os bloqueadores veiculares e rastreadores são uma alternativa para minimizar os prejuízos, tomando ações após o ladrão ter posse do veículo, tornando possível a recuperação do mesmo rapidamente.

Geralmente, o número de furtos veiculares ou objetos internos é bem maior do que o número de roubos, pois o ladrão encontra maior facilidade e

menor perigo para a sua realização. Por isso o mercado de segurança veicular tem expandido rapidamente, sendo produzidos diversos dispositivos e alarmes para tentar impedir o acesso do suposto ladrão ao veículo, minimizando o número de sinistros e aumentando sua segurança.

Existem vários tipos de alarmes automotivos no mercado, cada um com suas determinadas funções, preços e público alvo. Atualmente, a maioria dos alarmes de baixo custo quando acionados emitem sons diversos que são escutados nas proximidades do veículo e quando os usuários estão por perto. No máximo podem inibir a ação do suspeito, caso o mesmo esteja despreparado. Porém, eles não garantem que o dono do veículo será avisado que o alarme foi acionado e, se o ladrão tiver conhecimento sobre o alarme pode até desativá-lo rapidamente.

O objetivo desse trabalho é desenvolver um sistema que realize a interação entre o alarme veicular e o usuário, alertando-o, mesmo estando longe do veículo, e possa tomar suas devidas providências de acordo com a informação transmitida pelo dispositivo. Além disso, ele pode ser acoplado juntamente a um alarme convencional, utilizando seus sensores e, caso seja possível, interagindo com o próprio alarme presente no veículo para uma maior eficiência, porém essa integração não é o foco desse projeto.

Após pesquisar sobre o tema, foram encontradas algumas patentes de projetos que utilizavam alarmes integrado a celulares e patenteados no Instituto Nacional da Propriedade Industrial - INPI, porém, sua divulgação ainda é pequena no Brasil e os produtos que chegam a ser vendidos não são de grande porte.

Também foram encontrados, em buscas pela internet, dispositivos que possuem funções semelhantes às apresentadas nesse projeto, mas a maioria deles utiliza celulares específicos que, após alguns anos, não são mais fabricados, sendo complicado para a sua reposição e evolução. Outro dispositivo de alerta pelo celular encontrado utilizava um módulo que possui apenas suporte ao envio de mensagens SMS (Short Message Service). Como esse tipo de mensagem pode ter atraso ou falhas, não é indicado a ser o único tipo de alerta.

Além disso, esse projeto não foi baseado nos projetos citados anteriormente, possuindo uma arquitetura diferente e módulos únicos.

Um dos objetivos do projeto é realizar um produto a baixo custo, tornando-o mais acessível à população, sem perder sua eficácia no alerta ao usuário e qualidade do produto final.

Para simplificar a utilização do projeto e tornar possível sua comercialização, o mesmo foi dividido em módulos, sendo obrigatório o Módulo básico que contém o circuito com microcontrolador, o módulo GSM (Global System for Mobile Communications), sensores de porta e de bateria do veículo. O módulo da bateria recarregável é opcional, mas altamente recomendado.

1.2 Visão Geral

Para realizar a integração entre o microcontrolador e os demais dispositivos serão conectados os componentes em um *protoboard*¹, sendo o

¹ *Protoboard* também conhecido por Matriz de Contatos ou Placa para Protótipos. Placa com diversos orifícios e conexões condutoras para montagem de circuitos temporários, sem o uso de soldas. Disponível em <http://www.feiradeciencias.com.br/sala15/15_02.asp>. Acessado em 20/09/2008

módulo GSM conectado ao circuito através das portas seriais TXD e RXD do microcontrolador.

A Figura 1 ilustra a interação entre o circuito e seus dispositivos em seu ambiente, sendo o módulo básico constituído do microcontrolador, módulo GSM, sensores presentes no veículo, que serão simulados no protótipo, e sensor de corte da bateria. Os demais sensores e o módulo de bateria recarregável são opcionais, porém recomendados para uma maior eficácia do sistema.



Figura 1 – Interação entre o Circuito e demais dispositivos.

(Fonte: o autor)

Ao acionar o protótipo, o microcontrolador se comunica com os sensores, que podem ser do próprio veículo ou adicionais, verificando se foram acionados. Quando algum sensor for acionado, o microcontrolador verifica a alteração de estado do pino onde o sensor está conectado e realiza as operações necessárias previamente configuradas. A principal operação realizada pelo microcontrolador, e foco deste projeto, é a comunicação com o celular do usuário através do módulo GSM, alertando sobre a possível tentativa de furto. A bateria recarregável será utilizada caso a bateria interna do veículo não forneça a voltagem necessária para o correto funcionamento do protótipo ou se ela for desconectada, deixando de alimentar o circuito do protótipo e o próprio veículo. Enquanto a bateria interna fornecer a voltagem necessária, ela também irá carregar a bateria recarregável.

1.3 Organização da Monografia

Capítulo 1 – Introdução: Neste capítulo estão apresentadas as idéias básicas sobre o projeto com os objetivos do mesmo e sua organização.

Capítulo 2 – Referencial Tecnológico: Este capítulo contém as informações tecnológicas utilizadas para o desenvolvimento e entendimento do projeto.

Capítulo 3 – Implementação do Projeto: Contém toda a parte de implementação do projeto, desde o projeto inicial até sua parte final, contendo as informações do microcontrolador, módulo GSM, da comunicação, entre outros detalhes.

Capítulo 4 – Resultados: Neste capítulo podem ser verificados os resultados sobre a realização do projeto, evidenciando a funcionalidade de acordo com os testes realizados.

Capítulo 5 – Conclusões: Aqui estão descritas as conclusões retiradas sobre o projeto desenvolvido, as dificuldades encontradas, provocando algumas mudanças para sua adaptação e as propostas sugeridas para a continuidade do projeto, sua ampliação e adaptação para tipos de utilização diversos.

2 Referencial Tecnológico

O conteúdo descrito no referencial tecnológico foi utilizado como base para a realização de todo projeto e é necessário para seu maior entendimento, explicando um pouco sobre os componentes utilizados.

2.1 *Alarme*

Existem diversos tipos de alarme no mercado, como os alarmes contra incêndio, furto, violações, temporais, eventuais, sensoriais, entre outros.

Sua funcionalidade em geral é para avisar ou sinalizar algum evento predeterminado, perigo ou erro, podendo haver uma resposta de acordo com o dispositivo ao qual o alarme está acionado. Alarmes com eventos predeterminados realizam automaticamente sua sinalização ao ocorrer o evento. Os alarmes manuais só realizam a sinalização ao ocorrer algum evento mecânico, como pressionar um botão, realizando a resposta determinada pelo dispositivo.

As principais sinalizações que podem ocorrer são sonoras e visuais. Os principais tipos de resposta utilizados pelos alarmes são de corte de energia, acionamento de circuito auxiliar ou bloqueio de determinado circuito, ligação, envio de mensagens predefinidas, abertura ou fechamento de portas, regulação de dispositivos, execução de uma operação predeterminada, entre outros.

2.1.1 Alarmes Veiculares

Há várias opções de alarmes veiculares, cada uma com a sua devida aplicação, operação e custo.

Os alarmes mais utilizados nos veículos de pequeno e médio porte são chamados de alarmes convencionais e geralmente possuem diversos tipos de sensores, que são acoplados à central do alarme e, quando disparados, emitem alertas sonoros e visuais tentando chamar a atenção de quem está em volta do veículo, alertar o dono sobre alguma tentativa de furto e assustar o suposto ladrão.

Os alarmes com rastreamento, que utilizam geralmente um dispositivo GPS para rastrear o veículo, informam as coordenadas geralmente pelo padrão NMEA-0183. padrão NMEA-0183 define os requisitos dos sinais elétricos, do protocolo de transmissão de dados e tempo e formatos específicos de sentenças para barramento de dados serial de 4800-baud². Esse padrão é projetado para suportar transmissão de dados serial unidirecional de um único transmissor para um ou mais receptores. Os dados são padronizados em ASCII e podem incluir informações como posição, velocidade, profundidade, etc. [NMEA, 2009]

Em geral, os alarmes com rastreamento são disponibilizados por empresas prestadoras de serviço em segurança veicular que realizam o monitoramento do veículo, podendo disponibilizá-lo também ao cliente para maior comodidade. Porém, sua utilização é restrita geralmente aos veículos caros ou

² *Baud* é uma medida para mudanças de sinal por segundo. Disponível em <<http://www.56k.com/glossary.shtml>>. Acessado em 04/08/2009

que possuem bens de alto valor, pois os valores dos planos para sua aquisição são bem mais elevados do que as dos alarmes convencionais.

Os bloqueadores veiculares são dispositivos geralmente silenciosos que realizam o bloqueio de partes determinadas do carro, como motor, combustível e volante, podendo ser realizados automaticamente, ao ser acionado algum evento previamente configurado, ou através de dispositivos remotos. Seu custo depende das funções realizadas. Quanto maior sua funcionalidade e abrangência, maior o seu custo, mas geralmente são mais acessíveis que os alarmes com rastreamento, visto que não há mensalidade para sua utilização. Porém, eles também podem ser utilizados pelas próprias prestadoras de serviços juntamente com os de rastreamento para realizar bloqueios específicos do carro.

Assim sendo, o foco desse projeto é alertar o usuário sobre o acionamento de algum sensor do veículo. Por isso, ele possui uma grande semelhança com o funcionamento e objetivos dos alarmes convencionais. Porém, como o projeto utiliza a rede GSM para enviar um alerta ao celular do usuário ao invés de alertas sonoros e visuais no próprio veículo, ele possui uma maior eficácia em sua principal função, possibilitando uma reação do usuário mais rápida e eficiente. Além disso, ele pode ser adaptado para enviar as coordenadas do carro ao usuário através de um módulo GPS, possibilitando o seu rastreamento, utilizar equipamentos que bloqueiam circuitos específicos do veículo e utilizar dispositivos que emitam alertas sonoros e/ou visuais a fim de intimidar o suposto ladrão. Lembrando que essas funções extras não são o foco desse projeto, podendo ser realizadas como propostas futuras ou adaptadas pelo próprio usuário.

2.2 Sensores Veiculares

O sensor mais básico dos alarmes veiculares é o sensor de porta, que geralmente já vem com os veículos, pois é o mesmo utilizado para acender a luz no interior do veículo ou avisar quando alguma porta se abre. Sua funcionalidade se dá geralmente através de sensores de pressão, assim, quando a porta está fechada o sensor está pressionado, mantendo o circuito fechado. Ao abrir o circuito é enviado um sinal que dispara o alarme, caso esteja programado. [HOWSTUFFWORKS, 2009]

A figura 2 ilustra um tipo de sensor de porta veicular presente na maioria dos veículos atualmente.

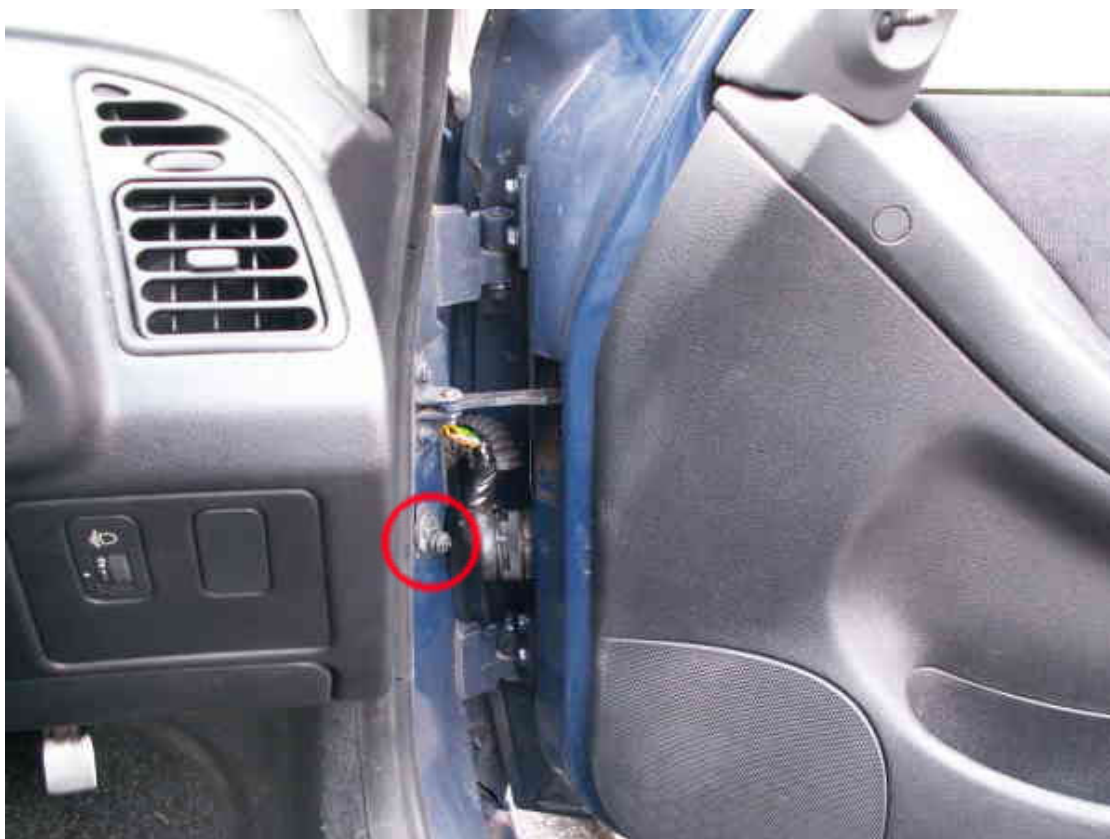


Figura 2 – Sensor de porta veicular

(Fonte: http://www.dantas.capugsensor_porta.jpg Acessado em 02/11/09)

Outro tipo de sensor utilizado é o sensor de ultra-som. Esse sensor é constituído basicamente de um emissor de ultra-som e um receptor. O emissor emite ondas sonoras, que se propagam a uma velocidade de, aproximadamente, 344 m/s, que são refletidas por obstáculos internos do veículo e são captadas pelo receptor. Os sensores de ultra-som são utilizados pelos alarmes convencionais para disparar quando houver alguma alteração de posição no padrão inicial registrado ao ser acionado, pois a distância percorrida pela onda sonora é menor quando algum objeto se encontra entre os obstáculos inicialmente registrados e o sensor, ou maior quando algum obstáculo é retirado ou movido. A precisão do sensor é diferenciada, de acordo com sua utilidade, pelo próprio fabricante. [HOWSTUFFWORKS, 2009]

Pode-se verificar na Figura 3 o funcionamento básico de um sensor ultra-som, que possui um receptor e um transmissor conectados no mesmo circuito.

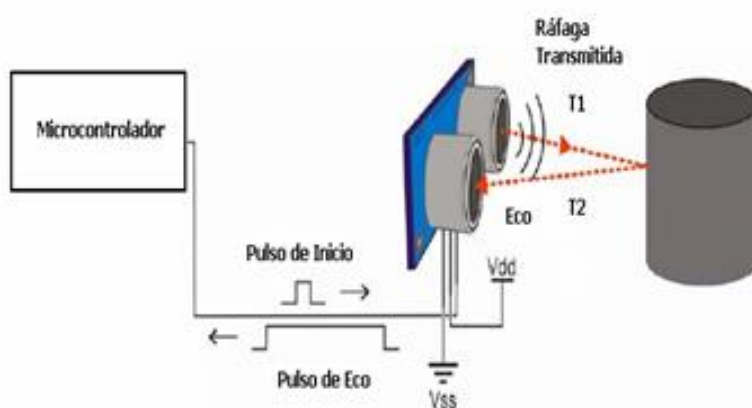


Figura 3 – Funcionamento do sensor ultra-som com receptor e transmissor.

(Fonte: http://www.rosoccontrol.com/Espanol/iBOARD/170_iBOARD_Ping_IR/PING/PingIR.htm Acessado em 18/10/08)

Existem também os sensores de choque, cuja função é enviar um sinal quando ocorrer algum movimento no veículo ou objeto onde o sensor se encontra, indicando sua intensidade. Esse sinal deve ser interpretado e pode enviar um alerta de acordo com a intensidade do movimento realizado. Esses sensores podem ser mais simples, sem diferenciar a intensidade do choque, podendo ocasionar erros fechando o circuito em choques pequenos, ou robustos, onde a intensidade do choque é diferenciada através de diversos contatos elétricos, dependendo do circuito utilizado. [HOWSTUFFWORKS, 2009]

A figura 4 ilustra um sensor de choque. No caso dos veículos, os locais mais prováveis para inseri-lo seriam em ambos os pára-choques.



Figura 4 – Sensor de choque com acelerômetro

(Fonte:

http://www.pcb.com/contentstore/images/PCB_Corporate/Vibration/products/photo/400/3501A1220KG.jpg Acessado em 03/11/09)

Outro sensor bastante utilizado é o sensor de janela, ou sensor que mede a pressão interna do veículo. Os mais comuns utilizados em veículos possuem um microfone, que mede as variações da pressão do ar e converte em uma corrente elétrica. A quebra de vidro possui sua própria frequência, assim a corrente elétrica captada pelo microfone é filtrada por um dispositivo, geralmente uma combinação de indutores e capacitores, que conduz apenas essa corrente

específica com a frequência da quebra do vidro, fechando o circuito.
[HOWSTUFFWORKS, 2009]

Podem-se utilizar também alto-falantes como sensores para verificar alterações na pressão do ar, através do eletromagnetismo, que ocorrem ao forçar uma janela, abrir uma porta, quebrar um vidro, entre outras atividades realizadas quando o suposto ladrão acessar o interior do veículo, gerando uma corrente elétrica. Um dispositivo deve gerenciar essa alteração da corrente elétrica para acionar os alertas a partir de certa alteração definida pelo mesmo.
[HOWSTUFFWORKS, 2009]

A figura 5 ilustra um alto-falante com os principais componentes utilizados em sua composição.

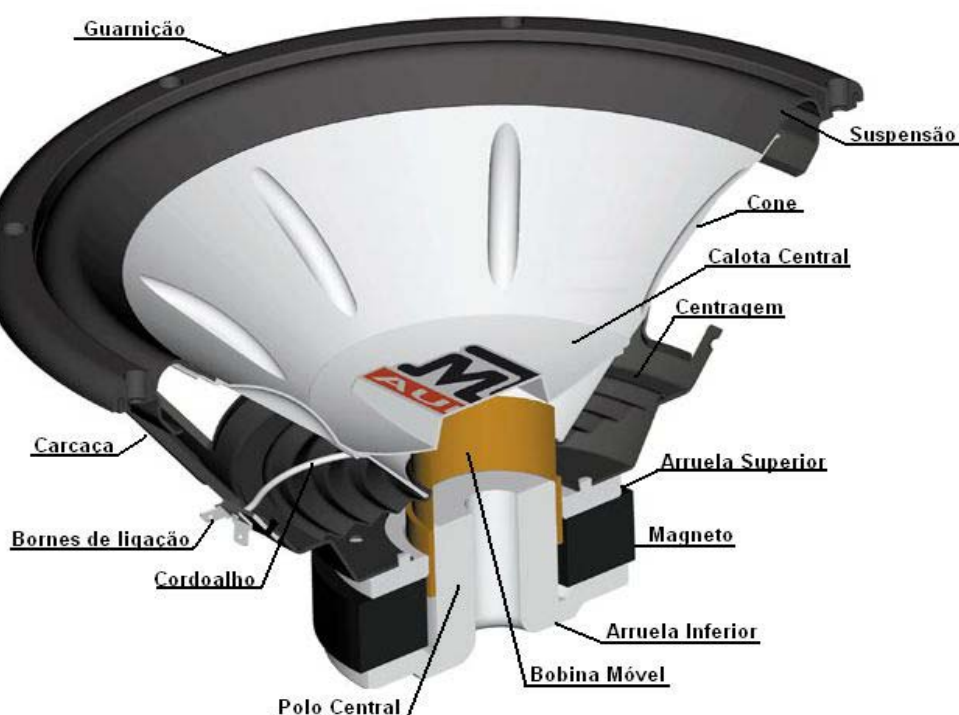


Figura 5 – Vista geral de um alto-falante e seus componentes

(Fonte:

http://autosom.net/artigos_user/artigo_20060926142437ArtigoBobinasRev.pdf

Acessado em 04/11/09)

Após pesquisar sobre os sensores utilizados em veículos, foi verificado que, atualmente, grande parte dos veículos é fabricada com alguns sensores, principalmente os de porta. Os mais específicos, como os sensores de pressão e de choque, não são facilmente adquiridos em pequena escala, tornando seu custo elevado.

Assim sendo, foi adotada a alternativa de apenas simular tais sensores através de chaves *pushbutton*³ sem trava. Portanto, enquanto o *pushbutton* sem trava não é pressionado o circuito onde ele é inserido permanece aberto, não conectando os seus terminais, como ocorre no funcionamento da maioria dos sensores automotivos. Quando ele é pressionado, seus terminais são conectados momentaneamente, simulando um sensor sendo acionado.

Foi escolhido o *pushbutton* sem trava, pois ele possibilita um funcionamento similar a alguns sensores que são acionados apenas por um curto espaço de tempo, fechando o circuito e abrindo logo em seguida, em um único acionamento. Isso não ocorre em chaves *pushbutton* com trava, pois essas fecham o circuito ao serem pressionadas uma vez e se abrem ao serem novamente pressionadas.

A figura 6 ilustra um exemplo de uma chave *pushbutton* sem trava de duas hastes para conectar dois terminais.

³ *Pushbutton* é um componente que conecta dois ou mais pontos do circuito quando pressionados. Disponível em <http://www.arduino.cc/en/Tutorial/Pushbutton>. Acessado em 12/04/2009.



Figura 6 – Chave *pushbutton* sem trava.

(Fonte: <http://www.reidosom.com.br/chavespushbutton.htm> Acessado em 06/04/09)

2.3 Prestadoras de Serviços Veiculares

Usuários que pretendem ter uma maior segurança sobre seu veículo muitas vezes necessitam contratar uma prestadora de serviços que utiliza dispositivos, como GPS, satélites, circuitos com celulares ou módulos específicos, para rastreá-lo e monitorá-lo, obtendo uma maior interação sobre o mesmo.

Geralmente, é necessário pagar pelos equipamentos utilizados no veículo do cliente fornecidos pela Prestadora de serviço para realizar sua localização e monitoramento. Esses equipamentos variam de acordo com a empresa contratada e plano adquirido, podendo haver outras funções. Além disso, eles possuem uma mensalidade por esses serviços, suporte, assistência e monitoramento.

Assim sendo, os aparelhos e a mensalidade necessária têm um custo elevado ao todo, não sendo viável à maioria da população residente no Brasil.

2.4 Microcontrolador

Microcontrolador é um circuito impresso com alta densidade de integração que inclui, dentro do chip, a maioria dos componentes necessários para o controlador. [ZELENOVSKY, 1999]

O microcontrolador corresponde a um microprocessador e seus periféricos, todos em um só chip⁴, onde geralmente estão inclusos o microprocessador, memória ROM e RAM, temporizadores/contadores, canais de comunicação, entre outros. [NICOLOSI, 2004]

A CPU⁵ dos microcontroladores são menos poderosas do que a dos microprocessadores, limitando seu conjunto de instruções, sendo assim mais simples. Além disso, em comparação com os microprocessadores, sua frequência de processamento, ou *clock*, é mais baixa e seu espaço de memória endereçado também costuma ser bem menor, sendo aplicado a sistemas com menor complexidade e menor custo. [MOHR, 2001]

A idéia básica do microcontrolador é a de construir um processador, com uma grande quantidade de recursos voltados para o controle. Com isso, o sistema fica muito mais simples e emprega poucos componentes. [ZELENOVSKY, 2002]

Existem diversos tipos de microcontroladores no mercado. As principais características que os diferencia são a quantidade de memória interna, velocidade de processamento, quantidade de pinos de entrada e saída, voltagem de alimentação suportada, pinos com funções específicas, arquitetura, entre outras.

Após análise de vários tipos de microcontroladores, foi escolhido um microcontrolador da família 8052, uma variação da família 8051 que possui a maioria dos circuitos e componentes idênticos, com algumas melhorias e

⁴ *Chip* também chamado de Circuito Integrado. (NICOLOSI, 2004)

⁵ *CPU* é a Unidade Central de Processamento, o “cérebro” do sistema. (NICOLOSI, 2004)

evoluções. Um dos motivos da escolha foi devido a sua ampla utilização comercial, sendo a família 8051 estudada na disciplina “Microprocessadores e Microcontroladores (com Laboratório)” realizada no curso de Engenharia da Computação.

A família dos microcontroladores 8051 foi criada no início da década de 80, possuindo 8 bits, sendo herdeira do 8048. Tipicamente, os microcontroladores da família 8051 possuíam internamente uma RAM de uso geral de 128 bytes e 128 bytes correspondentes aos registradores especiais, ROM de 4KB, quatro portas de entrada e saída, dois Timers de 16 bits, uma interface serial, capacidade de 64KB de endereçamento externo de ROM e de RAM, entradas de interrupção externa, entre outros. [NICOLOSI, 2004]

A família 8052 possui praticamente todos os componentes da 8051, com as seguintes melhorias: 8KB de memória ROM interna, um terceiro timer e 256 bytes de memória RAM interna para uso geral. [MOHR, 2001]

Para a realização do circuito do microcontrolador era planejado utilizar um kit didático pronto que tivesse um microcontrolador da família 8052. Porém, esses kits possuem limitações sobre a voltagem utilizada no circuito, número de portas disponíveis, expansão da placa para utilização de outros dispositivos, além de componentes que não seriam utilizados nesse projeto e ocupam pinos que poderiam ser utilizados para outras funções. Por isso, foi escolhido criar um circuito próprio, seguindo Kits de Estudo sobre microcontroladores e adaptando-os ao desejado, utilizando apenas os circuitos necessários.

Para o microcontrolador funcionar ele precisa de diversos componentes básicos. Os principais são: circuito de reset, não é obrigatório, mas

muito recomendado, circuito do oscilador com cristal, geralmente de quartzo, e circuito de alimentação, caso a voltagem fornecida não seja adequada ao microcontrolador.

O kit de estudo que foi utilizado como base é o kit de estudo 8051, que foi retirado do site EPOPÉIA, utilizando alguns conhecimentos, melhorias no circuito e complementos analisados no livro do autor ZELENOVSKY, ambos referenciados na bibliografia.

Dentre os microcontroladores da família 8052 foi escolhido o Atmel AT89S52, sendo esse um dos utilizados no projeto do kit didático adotado. Um dos principais motivos que levou a escolha deste microcontrolador é a possibilidade de fazer o próprio circuito gravar o programa na memória flash interna através da programação ISP⁶, sem precisar de uma gravadora específica. Além disso, ele permite sua programação em linguagem C, aceita um cristal de até 33MHz, possui seis vetores de interrupção, sendo três para os *Timers* 0, 1 e 2, duas para as interrupções externas INT0 e INT1 e uma para a comunicação serial, e dois modos para baixa utilização da CPU. [AT89S52, 2005]

A Figura 7 ilustra o microcontrolador utilizado nesse projeto.

⁶ *ISP*, sigla de In-System Programming, método utilizado para programar e reprogramar um microcontrolador posicionado dentro de um sistema ou circuito. Disponível em <<http://www.xeltek.com/pages.php?pageid=8>>. Acessado em 10/10/2008.

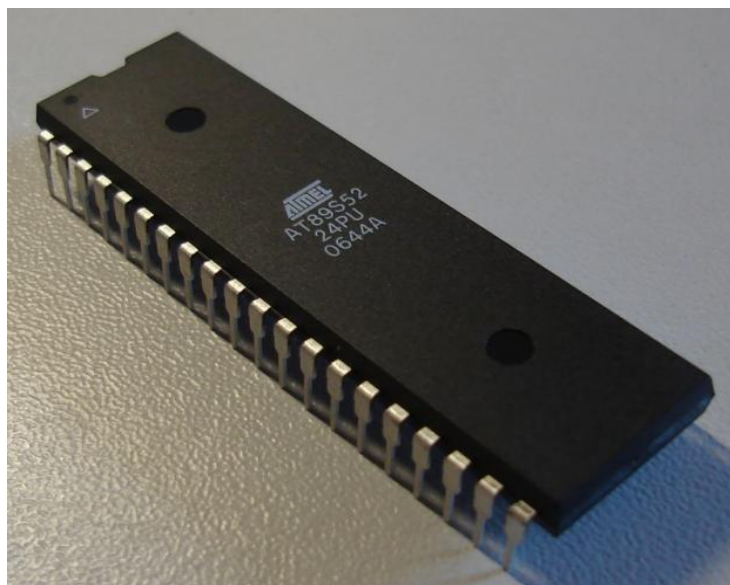


Figura 7 – Microcontrolador Atmel AT89S52

(Fonte: http://images01.olx.com.br/ui/3/76/52/44427452_2.jpg)

Acessado em 12/09/09)

2.4.1 Circuito do kit de estudo

Foi realizada, de acordo com o kit de estudo, a integração do circuito de *clock* com as especificações dos capacitores necessários recomendados pelo próprio *datasheet* do microcontrolador, parte do circuito para regular a fonte de saída para o microcontrolador e o circuito de *reset*, para reinicializar o programa do microcontrolador através de uma chave *pushbutton* sem trava. Foi utilizado também o esquema de *LED* (Light Emitting Diode), para verificação da ocorrência dos eventos gerados, da comunicação com o módulo GSM e dos *pushbuttons*, para a simulação dos sensores.

Além disso, ele também possui um visor LCD que não foi inserido ao projeto por não ser necessário, visto que o ambiente do projeto é no interior do veículo, em um local que não seja de fácil acesso, não sendo realizado para ser

verificado internamente, além de, geralmente, ocupar todos os pinos de uma porta para seu funcionamento.

As modificações realizadas sobre o kit foram a alteração da tensão de entrada para 12 V para a adaptação de uma única fonte para todos os circuitos utilizados, simulando a bateria do veículo, circuitos para adaptar a voltagem de saída para a comunicação entre o microcontrolador e módulo GSM, inserção do módulo da bateria recarregável com seus devidos componentes para seu perfeito funcionamento, a interface de conexão da porta serial do microcontrolador e a adaptação para a conexão do módulo GSM.

O kit didático disponibilizado possui ainda um circuito para gravação do programa ao microcontrolador pela porta paralela do computador, através do conector DB-25 macho. Essa conexão é mais simples e exige menos componentes para sua utilização comparando a gravação pela porta serial através do conector DB-9.

A Tabela 1 representa a seqüência dos pinos utilizados para gravação entre o computador e o microcontrolador.

Tabela 1 – Pinos de gravação pela porta paralela para o Microcontrolador.

(Adaptado de: EPOPEIA, 2008)

DB25M Computador	Nome	Referência	Pino AT89S52 Microcontrolador
6	RESET	RST	9
7	MOSI	P1.5	6
8	SCK	P1.7	8
10	MISO	P1.6	7
25	GND	GND	20

Porém, esse não foi utilizado, pois a gravação pela porta paralela não é totalmente segura, o computador utilizado apresentou problemas pela porta paralela e já havia um kit didático disponível para realizar a gravação pela porta serial, através do conector DB-9, com o circuito de proteção, dando maior segurança ao microcontrolador e à porta serial do computador.

2.4.2 Portas de Entrada e Saída

Cada microcontrolador possui um determinado número de portas de entrada e saída para comunicação externa. A família 8052, assim como a 8051, possui quatro portas de 8 bits cada, totalizando 32 pinos de entrada e saída, que podem ser utilizados e configurados de várias maneiras, sendo alguns já preestabelecidos pelo próprio fabricante para algumas funções específicas internas ao microcontrolador. [MOHR, 2001]

Porém, ao se utilizar algum pino com função interna específica, os outros pinos da mesma porta utilizada, que possuem funções internas, não podem ser utilizados como via de entrada e saída, mesmo que sua função não seja utilizada, podendo comprometer alguns pinos. [MOHR, 2001]

A vantagem de utilizar um microcontrolador com várias portas é que você pode ampliar o projeto com outros módulos, expandindo-o, sem ser necessário sua troca e, dependendo do novo microcontrolador a ser utilizado, adaptar todo o circuito.

A Figura 8 ilustra a disposição dos pinos do microcontrolador utilizado.

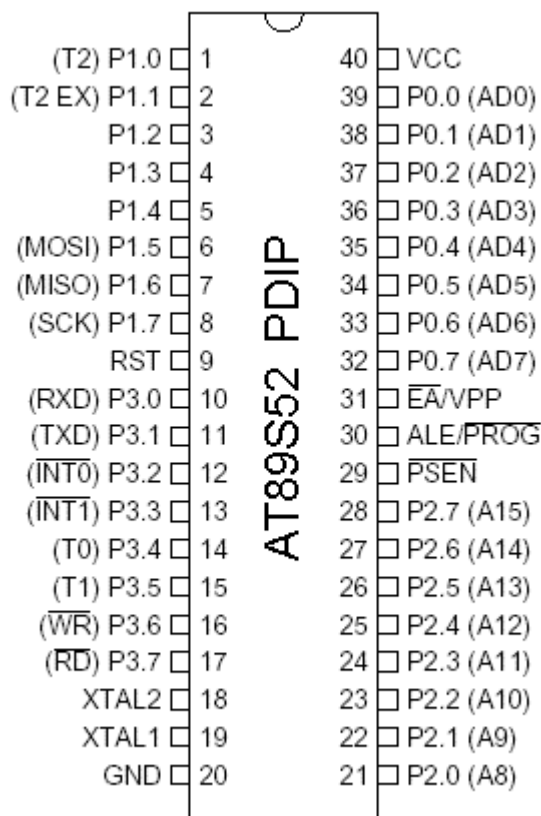


Figura 8 – Microcontrolador AT89S52

(Adaptado do anexo: Datasheet AT89S52)

Os pinos de entrada e saída desse microcontrolador estão na configuração *pull-up*. Nessa configuração, o microcontrolador entende que o pino está ativado quando o valor no pino for de 0 V e desativado quando o seu valor for de 1 V. Isso ocorre devido às características do microcontrolador de drenar corrente ao invés de fornecer. [EPOPÉIA, 2008]

Como podemos ver, a porta P1 do microcontrolador possui dois pinos específicos para o controle de timer e contador (T2 e T2 EX) e três pinos para programação ISP (MOSI, MISO e SCK), restringindo seu uso às funções internas, sendo disponíveis apenas os pinos 2,3 e 4 dessa porta para uso geral.

Os pinos da porta P3 estão relacionados a dois pinos para comunicação serial (RXD e TXD), dois para interrupções externas (INT0 e INT1),

dois para temporizadores ou contadores externos (T0 e T1), um para leitura (RD) e outro para escrita em memória externa (WR). Portanto, como será utilizada a comunicação serial e foram reservados os pinos de acesso à memória externa para possíveis operações futuras, não foi utilizado nenhum pino dessa porta para a utilização de sensores ou demais circuitos.

Pelas especificações do próprio microcontrolador utilizado, as portas P0 e P2 não possuem nenhum pino com função interna específica. Assim, a porta P0 é utilizada, inicialmente, para a simulação do acionamento de sensores através das chaves *pushbutton*.

Já a porta P2 é utilizada nesse protótipo para verificar, através de *LEDs*, o disparo de algum sensor e as etapas da comunicação entre o microcontrolador e o módulo GSM. Os *LEDs* não seriam necessários para uma proposta comercial, sendo utilizados apenas para uma fácil verificação na apresentação do protótipo.

Inicialmente, foi reservada para utilização dos sensores uma porta, podendo conectar até oito sensores no projeto. Caso seja necessário para propostas futuras, pode-se utilizar outra porta para o mesmo, verificando sua disponibilidade de acordo com os dispositivos e módulos integrados ao projeto.

2.4.3 Níveis de tensão

O microcontrolador utilizado possui níveis de tensão em seus pinos no padrão TTL⁷.

⁷ *TTL*, sigla de Transistor-Transistor Logic. São dispositivos digitais que trabalham com níveis de tensão definidos, onde o nível lógico baixo equivale a 0V e o alto equivale a 5V, aproximadamente,

A Tabela 2 apresenta os valores aproximados dos níveis de tensão de acordo com os pinos utilizados, sendo a tensão de entrada fornecida ao microcontrolador, através do pino VCC, de 5,1 V, aproximadamente.

Tabela 2 – Níveis de tensão de entrada e saída
(Adaptado do anexo: Datasheet AT89S52)

Símbolo	Parâmetros	Condições dos pinos	Min	Max
V_{IL}	Input Low Voltage	(Except EA)	-0.5 V	0,92 V
V_{IL1}	Input Low Voltage	(EA)	-0.5 V	0,72 V
V_{IH}	Input High Voltage	(Except XTAL1, RST)	1,92 V	5,6 V
V_{IH1}	Input High Voltage	(XTAL1, RST)	3,57 V	5,6 V
V_{OL}	Output Low Voltage	–	–	0,45 V
V_{OH}	Output High Voltage	–	2,4 V	–

2.4.4 Interface Serial

A família de microcontroladores 8051, assim como seus similares, possui uma USART (interface universal que permite a comunicação serial no modo síncrono ou assíncrono), permitindo um fácil acesso de leitura e escrita de dados à porta serial de outro dispositivo. [STEINER, 2005]

Na transmissão serial, podem ocorrer basicamente duas formas de comunicação: o síncrono e assíncrono. O síncrono realiza não apenas o transporte serial dos bits de dados, mas também caracteres de sincronismo, que

variando de acordo com a tensão de entrada fornecida para o dispositivo. Adaptado de <<http://books.google.com.br/books?id=YZqGTLckjLAC>>. Acessado em 06/11/2009.

ao serem detectados no receptor, ajustam os seus *clocks* internos para receberem os bits de dados na taxa de comunicação estabelecida. [COMUNICAÇÃO SERIAL, 2009]

Assim sendo, no modo síncrono, o pino de recepção serve tanto para transmitir como para receber e o pino de transmissão é utilizado como *clock* de referência, com a mesma frequência e fase para o pino de recepção.

Já na comunicação assíncrona cada caractere é transmitido individualmente, e para cada um têm-se os bits de início e os bits de parada. As taxas de comunicação devem ser previamente definidas entre os dispositivos que se comunicam. [COMUNICAÇÃO SERIAL, 2009]

Para a comunicação entre o microcontrolador e os dispositivos a ele conectados foi adotado o modo assíncrono, por ter menor custo em circuitos para a sua realização e a maioria dos dispositivos que se comunicam pela porta serial possuem suporte à comunicação assíncrona.

2.5 Terminal e módulo wireless GSM

Um Terminal GSM é um dispositivo projetado com capacidade de comunicação entre dispositivos, desenvolvido para aplicações M2M. Ele utiliza um módulo wireless GSM como circuito principal de processamento e demais circuitos para comunicação, visor LCD, entre outros.

Esse terminal geralmente é utilizado em manutenções remotas, transporte, logística, sistemas de tráfego, gateways celulares, vending machines, sistemas de segurança e de saúde, entre outros. [SIEMENS, 2008]

A pesquisa inicial sobre o Terminal GSM não trouxe muitos resultados, pois a maioria deles possui um custo elevado, tornando-se inviável para o projeto. O principal motivo do alto custo dos terminais é por causa da integração do mesmo com o módulo GSM, sendo necessários diversos componentes, como microfone, alto-falante, fonte externa e display LCD, para permitir a integração total com todos os circuitos do módulo, habilitando todas suas funcionalidades. Para a realização desse projeto é necessário apenas algumas de suas funcionalidades.

Portanto, foi realizada uma pesquisa sobre módulos que podem ser utilizados sem o seu terminal, sendo necessária a realização dos circuitos para seu correto funcionamento a parte. Como resultados dessa pesquisa, foram encontrados alguns tipos de módulo GSM que suprem todas as funcionalidades requeridas pelo projeto e possuem um ótimo custo/benefício.

Uma grande vantagem desses módulos é que a maioria deles possui o mesmo padrão de comunicação serial, níveis de tensão e voltagem do circuito, além de utilizarem os comandos AT. Portanto, caso a fabricação do módulo GSM utilizado nesse projeto seja descontinuada pelo seu fabricante ou haja algum outro módulo de qualidade superior, com maior utilidade e melhor custo/benefício; a troca desse componente pode ser realizada com apenas algumas adaptações físicas, se necessário, já que o padrão dos comandos utilizados é o mesmo. Assim sendo, torna-se desnecessário alterar o código do programa, compilá-lo novamente no microcontrolador e trocar de diversos circuitos para a troca de um módulo com os mesmos padrões.

Após diversas pesquisas e aquisições de módulos que não puderam ser adaptados para esse projeto, que são detalhados no capítulo 5, foi encontrado

o Módulo Wireless GSM/GPRS BenQ M22. Este módulo tem a vantagem de suportar a comunicação via GPRS (General Packet Radio Service), sendo possível uma transmissão de dados pela internet ou um envio mais seguro de mensagens SMS, caso seja corretamente configurado e a operadora utilizada possua suporte. Isso facilita para futuras evoluções no projeto. Além disso, ele possui esquemas de como criar o seu próprio terminal e realizar os circuitos desejados do módulo, não sendo necessária a compra de um terminal específico para sua integração.

A Figura 9 ilustra o circuito integrado com o módulo GSM utilizado no projeto que possui os componentes básicos para seu funcionamento e fios conectados aos pinos necessários do circuito que são utilizados para alimentação, controle e comunicação.

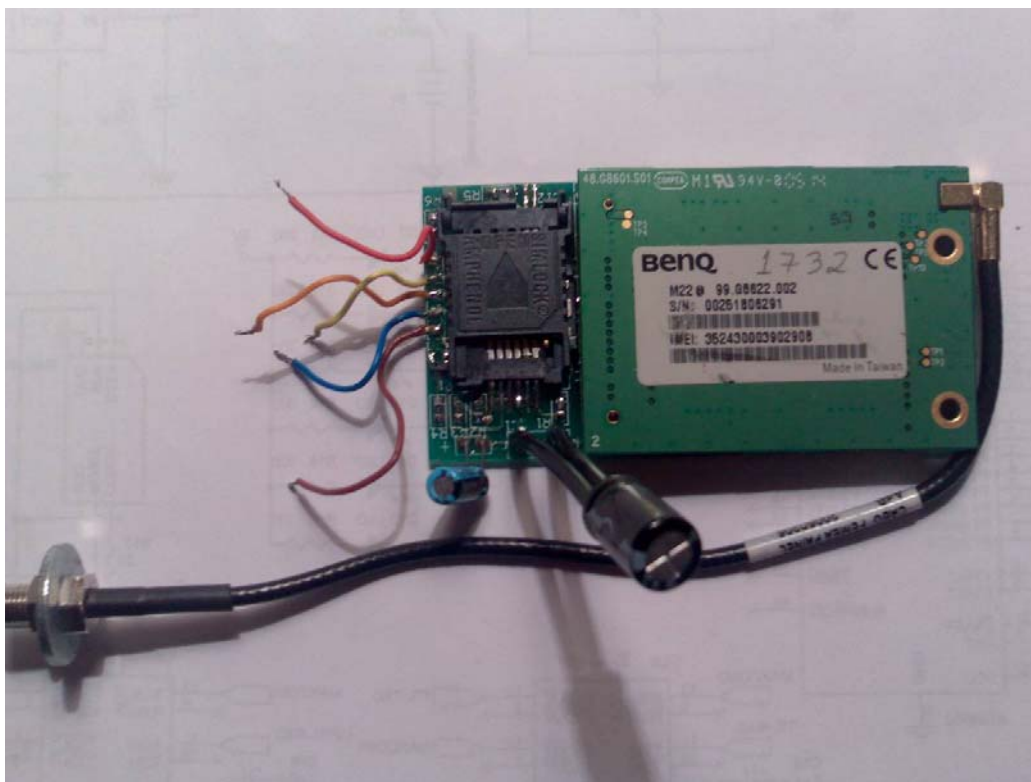


Figura 9 – Circuito integrado com Modem GSM/GPRS BenQ M22

(Fonte: o autor)

2.5.1 Níveis de tensão

O módulo utilizado possui níveis de tensão em seus pinos no padrão 3.3-V LVTTL⁸.

A Tabela 3 apresenta os valores aproximados dos níveis de tensão de acordo com os pinos utilizados e a tensão de entrada fornecida ao módulo GSM no protótipo de, aproximadamente, 4 V.

Tabela 3 – Níveis de Tensão dos pinos utilizados do módulo GSM

(Adaptado de: DATASHEET M22, 2004)

Name	Description	Parameter	Min	Max
RXD	RS232: receive data.	Voltage Input High	1.96 V	3.3 V
		Voltage Input Low	-0.5 V	0.84 V
TXD	RS232: transmit data.	Voltage Output High	2.24 V	–
		Voltage Output Low	–	0.616 V
PWON	POWER ON pin.	Voltage Input High	2.8 V	–
		Voltage Input Low	–	1.2 V

⁸ 3.3-V LVTTL, sigla de Low Voltage Transistor-to-Transistor Logic. É o padrão de níveis de tensão criado pela JEDEC (Joint Electronic Devices Engineering Council) para circuitos que operam a partir de 3V. Disponível em <<http://www.edn.com/article/CA339711.html>>. Acessado em 07/11/2009.

2.6 Comandos AT

Os comandos AT foram originados de uma linguagem específica de comandos para modems, chamada *Hayes Command Set*, que se tornou um padrão para controle de modems, entre outros dispositivos. A sigla AT é um mnemônico de *Attention*, que significa atenção em português, sendo utilizada como prefixo à maioria dos comandos utilizados. [PC MAGAZINE, 2008]

Esses são comandos geralmente utilizados por celulares e modems compatíveis para configurações, realização de chamadas, verificações de funcionalidades e atributos dos modems, envio de mensagens, ligações, entre outros.

Alguns dispositivos podem ter suas restrições a alguns comandos desse padrão. Os terminais e módulos wireless GSM possuem suporte a praticamente todos os comandos do padrão AT e, geralmente, possuem alguns comandos extras para funções específicas do módulo com o intuito de facilitar a sua utilização ao usuário.

2.7 Linguagem de Programação

Existem dois tipos principais de linguagens de programação: as linguagens de baixo nível, onde os comandos são executados diretamente pela máquina e as instruções e os dados são trabalhados em nível binário, e as linguagens de alto nível, que são independentes da arquitetura do hardware e melhor entendida por programadores, gerando facilidades na programação, porém essas devem ser compiladas para linguagem de máquina para sua execução. [NICOLOSI, 2004]

A linguagem adotada para o projeto foi a programação C. Uma das razões dessa escolha foi a quantidade de bibliotecas, funções e materiais de apoio disponibilizadas na internet sobre a programação em C, além de que a maioria dos microcontroladores encontrados atualmente no mercado já são integrados com compiladores da linguagem C para seu desenvolvimento.

3 Implementação do Projeto

Neste capítulo é detalhada a implementação do projeto, desde sua fase inicial até sua fase final, com o detalhamento de todo o circuito físico utilizado no protótipo, explicando as principais funções realizadas pelo programa do microcontrolador.

3.1 Adaptação em Protoboard

O protótipo foi adaptado do kit de estudo do site EPOPÉIA para ser construído em um *protoboard*, por ter uma facilidade e agilidade em seu manuseio, troca e inserção dos componentes utilizados. Além disso, como o intuito do projeto é o de simular os eventos que ocorrem quando os sensores são acionados, assim como o controle dos mesmos, não é necessário construir um circuito impresso para reduzir o seu tamanho.

3.2 Fonte

O projeto apresenta diversos dispositivos que necessitam de uma voltagem diferente para seu correto funcionamento. Como seu ambiente simula o interior de um veículo, a única fonte de tensão disponível é sua bateria, que possui 12 V de tensão e corrente contínua. Portanto, foi criado o circuito da fonte para adaptar a voltagem da bateria às demais necessárias.

A Figura 10 mostra parte do circuito da fonte. Este circuito foi montado e utilizado para recarregar a bateria recarregável com uma corrente baixa e

alimentar os demais circuitos. Os valores de todas as resistências abaixo estão em ohms (Ω).

Fonte 7,4 a 8,2 V

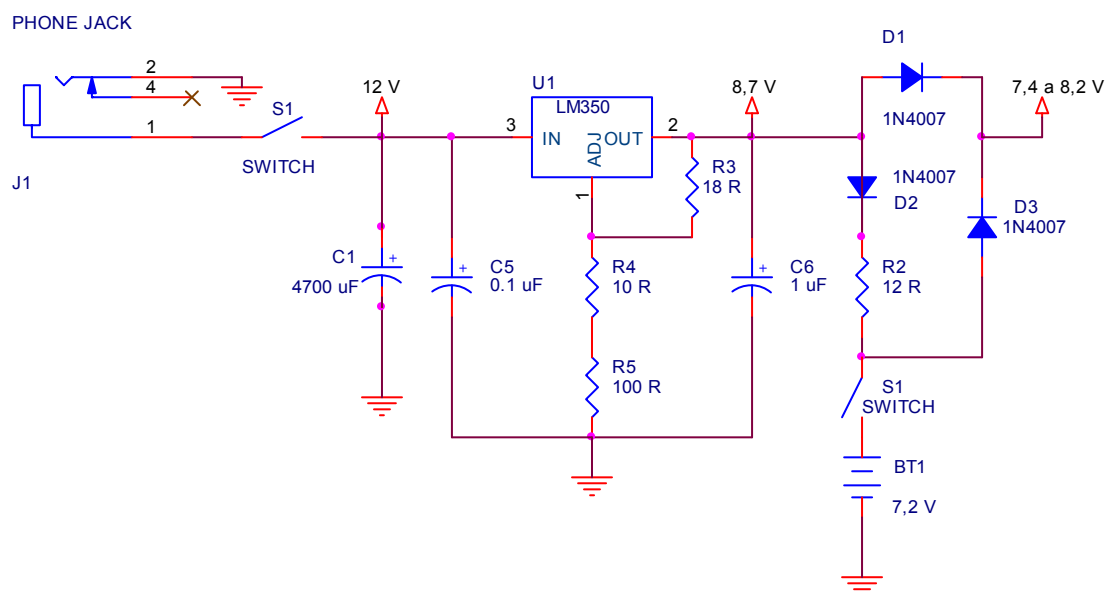


Figura 10 – Circuito da fonte que recarrega a bateria e alimenta o circuito
(Fonte: o autor)

A fonte utilizada para simular a bateria do carro é uma fonte de 12 VDC e permite uma corrente no circuito de até 3,2 A, sendo essa conectada ao conector J1. As setas indicadas em alguns pontos da Figura 3.1 são as medições da tensão nesses determinados pontos.

Foi inserido um capacitor de 4700 uF na saída da fonte. O intuito desse capacitor é de filtrar a tensão para permanecer constante até mesmo em casos de alto consumo do circuito. Foi utilizado o LM350 (U1) para regular a tensão de entrada de 12 V para, aproximadamente, 8,7V. Essa tensão é alterada através dos ajustes com resistores (R3 e R4), desde que a tensão de entrada, pino 2 do LM350, seja, pelo menos, 1,2 V a mais do que a tensão de saída, pino 3

do LM350. Quando a tensão de 8,7 V passa pelos diodos (D1 e D2) ela cai para, aproximadamente, 8,2 V. Foi escolhida essa tensão de 8,7 V, resultando em 8,2 V de entrada na bateria, pois quando a bateria está totalmente carregada, sua tensão máxima seria de 8,4 V. Para não haver uma sobrecarga na bateria e como o circuito, dentro do carro, ficaria recarregando a bateria por um longo tempo, quando a bateria chegar aos seus 8,2 V a corrente iria parar de carregá-la, visto que a diferença de potencial entre o circuito e a bateria é zero.

Os diodos D1 e D2 foram inseridos para que, quando a fonte estiver conectada, a corrente siga em direção à bateria e alimente o restante do circuito do protótipo e quando a fonte for cortada não haver refluxo para os componentes da fonte, direcionando a corrente da bateria apenas para os circuitos desejados. Já o diodo D3 é utilizado para que a bateria não alimente o circuito a menos que a diferença de tensão entre a bateria e a saída do diodo D1 seja maior que, aproximadamente, +0,6 V. Isso ocorre quando a fonte é cortada, tornando a tensão da fonte nesse ponto igual a 0V, ou se o circuito demandar uma corrente elevada por um determinado tempo que seja capaz de reduzir a tensão da fonte nesse ponto para um valor 0,6 V abaixo da tensão da bateria, havendo uma diferença de potencial necessária para passar a corrente da bateria pelo diodo D3.

Para alimentar o circuito do módulo GSM é necessário que a tensão nesse circuito seja de, no mínimo 3,3 V, como pode ser visto através de seu datasheet em anexo. Portanto, foi realizada a adaptação da voltagem necessária para o correto funcionamento do circuito através do regulador de tensão LM317T.

A Figura 11 ilustra o circuito referente à alimentação do módulo GSM.

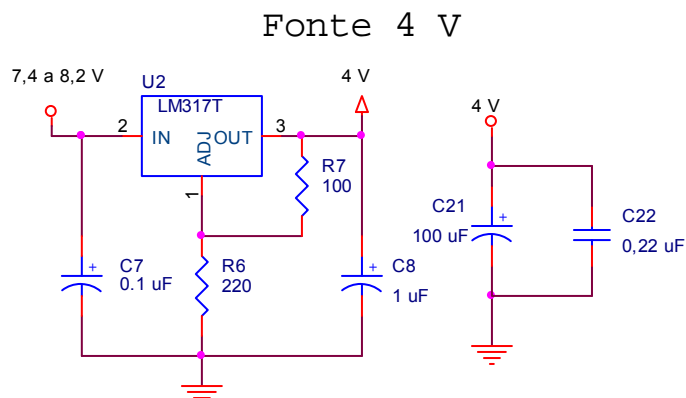


Figura 11 – Circuito da fonte para alimentação do Módulo GSM

(Fonte: o autor)

Esse circuito regula a tensão de saída no pino 3 do LM317 para 4V e é capaz de disponibilizar uma corrente até 1,5 A, podendo haver altos picos por um curto período. Essa é a tensão recomendada pelo fabricante do módulo GSM BenQ M22 para suprir, com sobra, a voltagem necessária para seu correto funcionamento. Os capacitores C7 e C8 são apenas para filtrar a tensão, mantendo-a constante e para aumentar sua impedância.

Já os capacitores C21 e C22 são utilizados para permitir picos de corrente necessários que os módulos GSM utilizam por um curto período em algumas operações específicas. O capacitor C21 é de Tântalo, enquanto o capacitor C22 é cerâmico. Esse circuito é recomendado por diversos manuais de módulos GSM e especialistas no assunto. O *datasheet* do BenQ M22 não comenta essa utilização, mas foi realizado mesmo assim para uma maior segurança.

Finalmente, para a alimentação do microcontrolador é necessária uma tensão de, no mínimo, 4 V, que foi adaptada utilizando o LM7805 para

disponibilizar 5V para o circuito, que é o recomendado pelo próprio fabricante e foram retiradas as recomendações do datasheet AT89S52 em anexo. Além disso o LM7805 é capaz de disponibilizar uma corrente constante de até 1 A.

A Figura 12 ilustra a parte do circuito referente à alimentação para o microcontrolador.

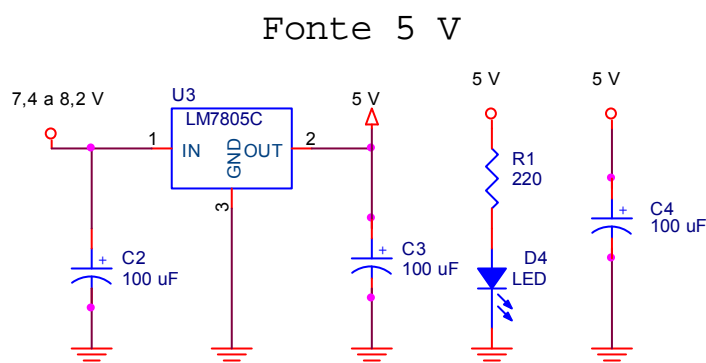


Figura 12 – Circuito da fonte para alimentação do microcontrolador

(Fonte: o autor)

3.2.1 Bateria recarregável

Foi analisado que para o correto funcionamento do circuito seria necessário uma bateria de aproximadamente 7V para garantir as funções básicas do mesmo, caso a bateria do veículo falhasse ou deixasse de fornecer energia. Para aperfeiçoar e automatizar o circuito, foi utilizada uma bateria recarregável para garantir o seu correto funcionamento, reduzindo a chance da bateria estar descarregada e do circuito ser desligado sem realizar a chamada ao usuário, alertando-o de algum possível evento, além de reduzir bastante as trocas de bateria.

Essa bateria recarregável foi criada através da união em série de seis pilhas recarregáveis de 1,2 volts e de 2800 mAh⁹, permitindo o circuito se manter ativo por um longo tempo antes de ser desligado, podendo o usuário ser alertado sobre os demais eventos ocorridos, além do corte da alimentação central. A opção das pilhas recarregáveis foi principalmente pela facilidade em sua substituição e comercialização, além de ser possível colocar pilhas de maior capacidade, caso sejam instalados outros módulos que necessitem de uma maior carga.

Inicialmente foi realizado um circuito que verificava se a alimentação estava correta e carregava a bateria através de um relé convencional. Assim sendo, quando a alimentação do circuito fosse “cortada” o relé iria redirecionar a alimentação para a bateria recarregável. Porém, como esse relé é um circuito eletromecânico, a troca de uma alimentação para a outra apresenta um alto retardo, reiniciando o microcontrolador e o módulo GSM, assim como todo circuito, prejudicando a comunicação com o usuário que poderia demorar ou, até mesmo, não ser realizada sem um devido tratamento.

Foi pesquisada a possibilidade de trocar por um relé de estado sólido, sendo seu retardo ínfimo comparado aos relés convencionais, mas isso iria complicar o circuito e torná-lo mais caro.

Para solucionar esse problema e facilitar sua solução, foi inserido um regulador LM317T, realizando o seu circuito básico, retirado do próprio datasheet do componente que se encontra em anexo na monografia, com resistores para

⁹ mAh, sigla para miliampére por hora. É a medida da capacidade de uma pilha ou bateria. Disponível em <<http://pilhas-recarregaveis.com.br/faq/influencia-capacidade-pilhas-nos-aparelhos>>. Acessado em 09/10/2008.

regular a tensão de saída para, aproximadamente, 8,2 V. Assim, quando a fonte, que simula a bateria do veículo, estiver ligada, sua voltagem de saída, além de ser regulada, será suficiente para carregar a bateria recarregável e alimentar o circuito e, ao ser desligada, a bateria recarregável passa a alimentar o circuito em um tempo desprezível, sem reiniciar os dispositivos do protótipo.

Essa utilização pode diminuir a vida útil das baterias, pois não possui um controle de carga e voltagem apurado, mas essa diminuição é pequena, principalmente por sua utilização não ser freqüente, e apresenta um alto custo/benefício e economia de circuitos.

3.3 *Microcontrolador*

Como dito anteriormente, o circuito do microcontrolador e os componentes básicos para seu correto funcionamento foram adaptados do kit didático disponível no site EPOPÉIA.

A Figura 13 ilustra o circuito adaptado para o microcontrolador, que é detalhado nos subitens a seguir.

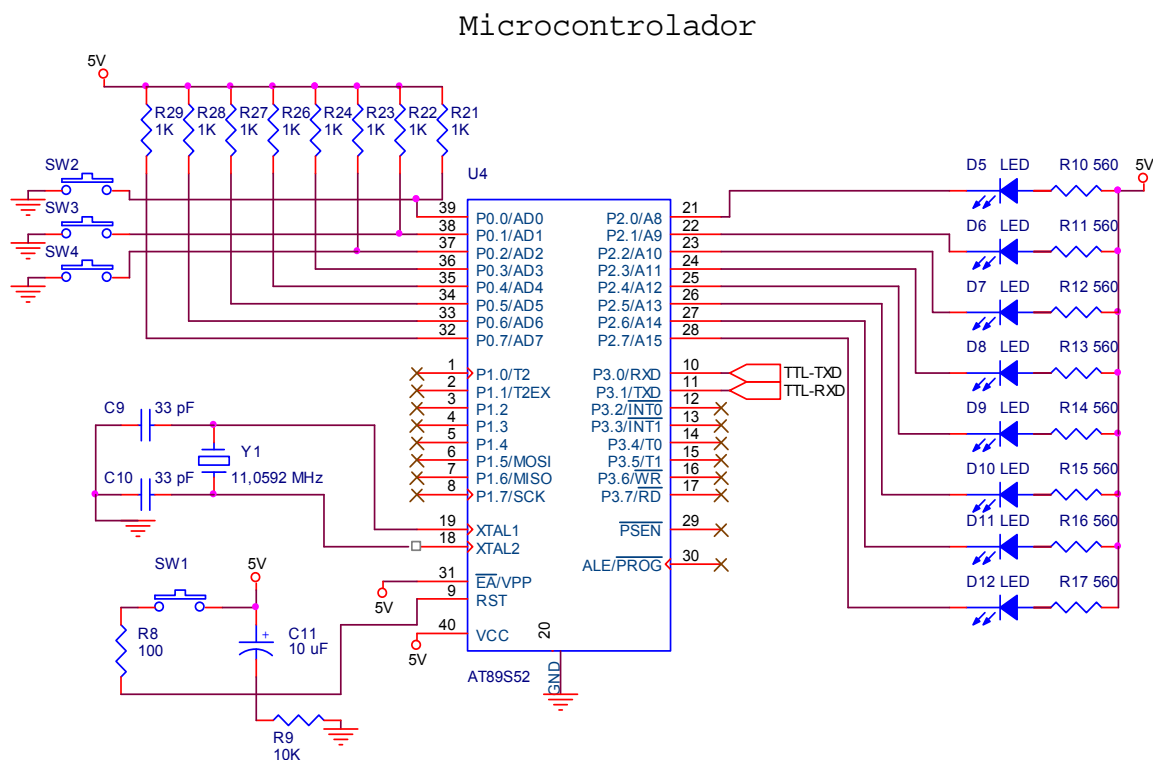


Figura 13 – Circuito do microcontrolador

(Fonte: o autor)

3.3.1 Oscilador

A frequência máxima de operação desse microcontrolador é de 33 MHz. O próprio datasheet do microcontrolador recomenda a utilização de um cristal de quartzo ou um ressonador cerâmico. Foi utilizado nesse projeto um oscilador de cristal de quartzo com a frequência de 11,0592 MHz (Y1), pois, além de ter uma larga utilização no mercado, esse valor permite trabalhar com comunicação serial em velocidades inteiras, reduzindo a taxa de erro. [STEINER, 2005]

A figura 14 mostra dois osciladores de quartzo, iguais ao utilizado no protótipo.



Figura 14 – Oscilador de quartzo de dois terminais

(Fonte: http://www.sobretom.com.br/49u_hfam.jpg Acessado em: 15/11/09)

Conforme o próprio manual do microcontrolador, torna-se necessária a utilização de dois capacitores cerâmicos, que podem variar entre 20 pF à 40 pF, para filtrar a frequência máxima do oscilador. Portanto, foram adotados os capacitores de 33 pF (C9 e C10) por recomendação e utilização no kit didático e em demais circuitos analisados.

3.3.2 LEDs

Os *LEDs* (D7 a D14) conectados ao microcontrolador são utilizados para verificar qual etapa do programa do microcontrolador está sendo realizada. Sua principal utilização é para fazer a verificação da comunicação entre o microcontrolador e o módulo GSM para saber se o módulo está realizando as operações corretamente de acordo com as solicitações do microcontrolador.

Cada *LED* está conectado a um determinado pino da porta P2. Como os pinos dessa porta estão configurados como *pull-up*, o microcontrolador drena a corrente do *LED* para acendê-lo. Pelo kit didático, os resistores (R11 a R18) utilizados para filtrar a corrente de entrada nos *LEDs* seriam de 330 ohms. Porém, essa resistência faria com que a corrente neles fosse alta, aumentando o seu brilho. Para minimizar essa corrente, com o intuito de aumentar o tempo de

funcionamento do circuito quando houver o corte da fonte principal e ele for alimentado pela bateria recarregável, foi alterado esse resistor para 560 ohms, reduzindo o seu brilho.

A Tabela 4 mostra qual etapa do programa cada *LED*, conectado ao seu pino, representa para uma fácil verificação das funções que estão sendo executadas pelo microcontrolador.

Tabela 4 – Etapas realizadas quando determinado *LED* acende
(Fonte: o autor)

Pino do Microcontrolador	Função realizada quando LED está aceso
P2.0	Alarme acionado pelo usuário e realizando a verificação nos sensores.
P2.1	Aguardando alguma resposta do módulo referente ao comando enviado.
P2.2	Esperando a resposta necessária para dar continuidade.
P2.3	Mensagem não enviada - Problema no envio (operadora ou sem sinal) ou sem crédito.
P2.4	Sensor Habilitado e realizando operações de resposta (enviando o(s) alerta(s)).
P2.5	O módulo está enviando dados ao microcontrolador.

3.3.3 Sensores simulados

Os sensores são simulados a partir de chaves *pushbutton* sem trava (SW2 a SW8). É utilizado esse tipo de chave, pois existem vários tipos de sensores que são acionados por um curto período, ao invés de ficar constantemente acionados, e depois voltam ao estado natural. Esse evento deve ser registrado como um único acionamento.

3.3.4 Reset

O circuito de reset foi realizado exatamente como o kit didático sugere. Utilizando um botão *pushbutton* sem trava (SW1), dois resistores, sendo um de 100 ohms (R9) e o outro de 10k ohms (R10), e um capacitor eletrolítico de 10 uF (C11). [EPOPÉIA, 2008]

3.3.5 Timer e Interrupção

O *Timer 0* está sendo utilizado como temporizador da interrupção para verificação do disparo dos sensores em ciclos predefinidos. Já o *Timer 1* é utilizado para gerar a taxa de transmissão de 9600-baud que é necessária para a comunicação entre o microcontrolador e o módulo GSM utilizando a porta serial. Por sua vez, o *Timer 2* é utilizado pela função *delay_ms*, para deixar o microcontrolador inoperante por um tempo estipulado. Foi utilizado o *Timer* para que esse tempo seja mais exato que utilizando apenas a programação para gerá-lo.

A Tabela 5 detalha o Registrador de interrupções do microcontrolador, sendo utilizada para habilitar as interrupções globais, configurando as do *Timer 0* e da porta serial, permitindo a verificação dos pinos e a comunicação dos dispositivos.

Tabela 5 – Registrador de Interrupções
(Adaptado do anexo: Datasheet AT89S52)

Símbolo	Posição	Função
EA	IE.7	Se EA=0, desabilita todas as interrupções. Se EA=1, as interrupções serão habilitadas ou desabilitadas de acordo com seu bit específico.
-	IE.6	Reservado.
ET2	IE.5	Bit para habilitar/desabilitar a interrupção do <i>Timer 2</i> .
-	IE.6	Reservado.
ES	IE.4	Bit para habilitar/desabilitar a interrupção da porta Serial.
ET1	IE.3	Bit para habilitar/desabilitar a interrupção do <i>Timer 1</i> .
EX1	IE.2	Bit para habilitar/desabilitar a interrupção externa 1.
ET0	IE.1	Bit para habilitar/desabilitar a interrupção do <i>Timer 0</i> .
EX0	IE.0	Bit para habilitar/desabilitar a interrupção externa 0.

3.4 Módulo Wireless GSM

O Módulo Wireless GSM precisa de alguns componentes para seu funcionamento. Como o foco do projeto é apenas alertar o usuário, muitos componentes do módulo não foram utilizados, podendo ser aproveitados para propostas futuras sobre o projeto. Um dos principais circuitos para seu funcionamento é o circuito do cartão *SIM* (Subscriber Identity Module), que possui o número do celular, um número próprio e único de identificação do cartão e

diversas configurações de serviços, como o centro de mensagem da operadora registrada e serviços exclusivos da mesma.

Na compra do módulo GSM já estava incluso um CI que fornece o circuito básico necessário para realizar a integração do cartão *SIM*. Além disso, ele disponibiliza fios conectados aos pinos de alimentação, terra e comunicação. Esse foi utilizado para o correto desempenho do módulo, simplificando o circuito que deveria ser realizado no protoboard.

A Figura 15 a seguir ilustra o circuito utilizado para o funcionamento do módulo GSM e sua comunicação com o microcontrolador.

Módulo GSM

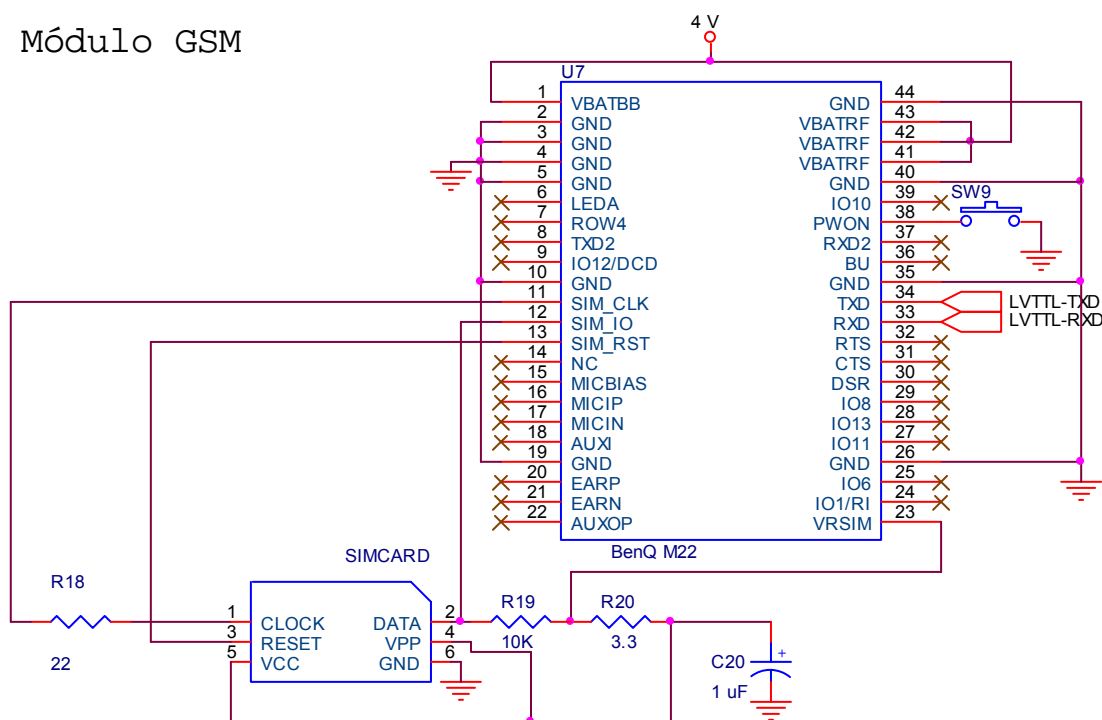


Figura 15 – Circuito do módulo wireless GSM

(Fonte: o autor)

Portanto, foi conectado ao circuito a tensão de 4 V através do fio de alimentação que conecta os pinos 1 (VBATBB), 41, 42 e 43 (VBATRF); o terra do

circuito através do fio que conecta os pinos 2, 3, 4, 5, 10, 19, 26, 35, 40 e 44 (GND) e os fios de comunicação, que estão conectados aos pinos 33 (RXD) e 34 (TXD), foram utilizados para a comunicação com o microcontrolador, que será detalhada no subitem 3.3.

3.5 Comunicação entre o Microcontrolador e o Módulo GSM

Para a comunicação entre o módulo GSM e o microcontrolador são utilizados dois circuitos MAX232N.

Isso ocorre, pois o módulo GSM possui o nível de tensão em seus pinos no padrão 3.3-V *LVTTL*, inclusive os pinos de comunicação, sendo a voltagem máxima permitida nos pinos de comunicação 3,3V. Maiores detalhes dos níveis de tensão dos pinos utilizados no módulo GSM podem ser analisados na Tabela 2.3, já apresentada no Capítulo 2. Já o módulo GSM trabalha com a tensão em seus pinos em nível *TTL*. Como a tensão mínima fornecida pelos seus pinos é de 2,4V, conforme descritos anteriormente na Tabela 2.2, é arriscado conectá-los diretamente ao módulo GSM, podendo queimar seus pinos em picos de tensão.

O componente MAX232, além de possibilitar a conversão dos diferentes níveis de tensão e impedâncias entre eles, possui também um circuito de proteção de descargas elétricas, reduzindo a possibilidade de queimar algum circuito devido a picos de descargas elétricas, conforme relata o datasheet do componente em anexo.

A Figura 16 ilustra o MAX232 e os pinos utilizados para a essa comunicação.

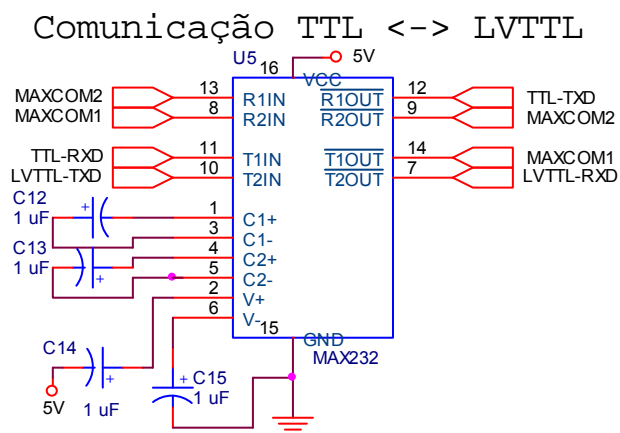


Figura 16 – Circuito para comunicação entre o microcontrolador e o módulo
(Fonte: o autor)

O circuito de capacitores e pinos utilizados para a comunicação foi adaptado de acordo com o datasheet do componente. Os pinos nomeados LVTTTL-TXD e LVTTTL-RXD são conectados aos pinos TXD e RXD do módulo GSM, respectivamente. Já os pinos TTL-RXD e TTL-TXD são conectados aos pinos RXD e TXD do microcontrolador. Por fim, os pinos MAXCOM1 e MAXCOM2 possuem a tensão nível RS232 e são adaptados para sua comunicação interna.

3.6 Software

O software foi desenvolvido em linguagem C, uma das suportadas pelo microcontrolador. Foi escolhida essa linguagem, pois foi a linguagem mais estudada no curso pelas disciplinas Linguagens e Técnicas de Programação I, Linguagens e Técnicas de Programação II e praticada também em Microprocessadores e Microcontroladores.

A figura 17 ilustra o fluxo operacional de funcionamento demonstrando os passos realizados pelo programa inserido no microcontrolador para facilitar o entendimento do programa.

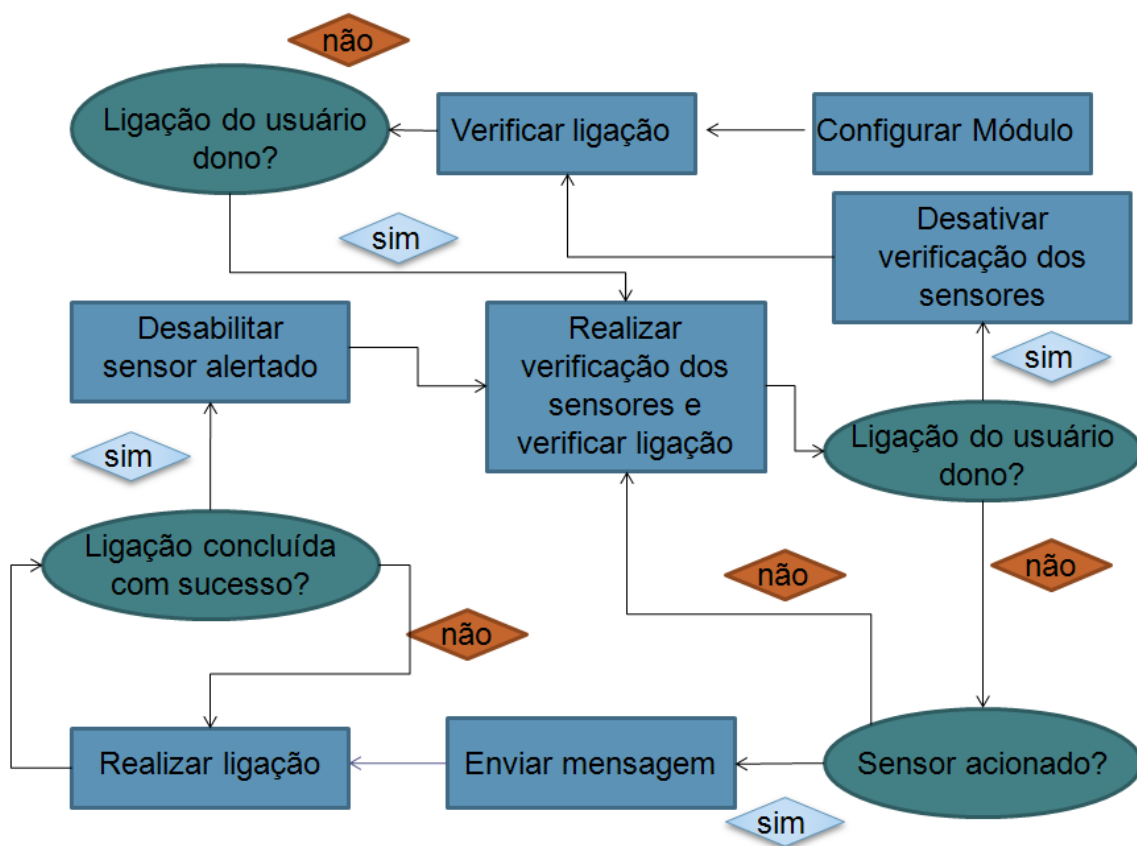


Figura 17 – Fluxo operacional de funcionamento

(Fonte: o autor)

As principais funções do software são para sincronizar o módulo GSM com o microcontrolador, padronizando a taxa necessária para sua comunicação, enviar comandos ao módulo, recebendo a sua resposta e verificando se é a resposta necessária para seu melhor funcionamento, tomar as medidas necessárias quando a resposta do módulo GSM não for a correta e verificar o estado dos sensores para saber quando algum foi acionado e, quando acionados, realizar o alerta ao usuário.

Foi escolhido que para o circuito ser acionado, monitorando os sensores e realizando suas respostas, é necessário que o usuário ligue para o protótipo e espere a ligação ser atendida. O circuito verifica se a ligação é do número cadastrado para receber os alertas e acionar o circuito, para não deixar que outra pessoa que possua o número do modem GSM desative-o.

A Figura 18 mostra o trecho do código que faz essa verificação. O número cadastrado no software que é autorizado a realizar o acionamento é o “84248528”, que está também cadastrado no cartão SIM utilizado com o nome “Dono”.

```

if (P2_5 != 1) // Faz a verificação se existe algum caractere que não foi lido na interrupção da porta serial.
{
    P2_1=0; //Aguardando Resposta do Módulo
    limpastring();
    do
    {
        while(P2_5!=0){};
        c=getchar();
        P2_5=1;
        if((c != '\0' && c != '\r' && c != '\n') )
        {
            retorno[t]=c;
            t++;
        }
    }while(c != '\r' && c != '\n' && t < 38 || t < 1);
    P2_1=1; //Resposta Recebida
    if (strstr(retorno,"Dono")!=NULL || strstr(retorno,"84248528")!=NULL)
    //Verificar se é o número cadastrado que está ligando
    {
        P2_1=1; //Resposta Recebida
        Aciona=1;
        Verificados=0xFF00;
        PortaAntes=0xFF00;
        P2_0=0;
        delay_ms(3000);
        reset_rx();
    }
}

```

Figura 18 – Código de acionamento para verificação dos sensores

(Fonte: o autor)

Independentemente se a resposta vier com o código DDD ou não ele irá ativar o circuito se o celular possuir o número descrito anteriormente.

A verificação dos sensores é realizada através de uma interrupção, utilizando o Timer 1, para que, mesmo quando o programa esteja executando alguma operação, se algum sensor for acionado ele verifica e armazena em uma variável para depois conferir se o alerta referente a esse sensor já foi realizado. Isso ocorre, pois o protótipo foi projetado para alertar apenas uma vez para cada sensor cadastrado, visto que existem diversos sensores que acionam várias vezes para apenas uma ocorrência, fazendo com que o usuário não seja alertado novamente por uma única ocorrência, tornando-o mais eficaz.

A Figura 19 ilustra o trecho do código referente a essa verificação do sensor através da interrupção e o código referente à espera do microcontrolador sem realizar operações que não sejam interrupções, definido em milissegundos (delay_ms).

```

void timer0_ISR () interrupt 1
{
    TRO=0; //Para a execução do Timer 0
    TMOD = ((TMOD & 0xF0)) | 0x01;
    PortaAtual=(~PD);
    if (((PortaAtual^PortaAntes)&PortaAtual)>=1)
    //verifica o estado da porta atual comparando com seu estado anterior
    {
        PortaAntes=(PortaAntes|PortaAtual);
        //Atribui o estado anterior ao atual para não verificar os pinos novamente
    }
    TRO = 1;
}

void delay_ms(unsigned int ms) //Função de geração de tempo - base 1us
{
    TL2 = 0x66; //Lower Byte
    TH2 = 0xFC; //Upper Byte
    RCAP2H = 0xFC; //Upper byte set value when reload occurs
    RCAP2L = 0x66; //Lower Byte set value when reload occurs

    TR2=1; //Habilita o timer 2.
    while(ms)
    {
        if (TF2==1) //Estouro do timer
        {
            ms--; // Decrementa a variável
            TF2 = 0;
        }
    }
    TR2=0; //Desabilita o timer 2.
}

```

Figura 19 – Código da função de atraso e da verificação dos sensores

(Fonte: o autor)

Caso um pushbutton que simula um sensor seja pressionado, ativando-o, e o circuito esteja acionado, o software enviará uma mensagem ao usuário, informando qual sensor foi acionado e verifica o status da mensagem. Caso a mensagem não seja entregue, ele acende o *LED* conectado ao pino 2.3 do microcontrolador. Como as causas de algum erro no envio da mensagem podem ser problemas na operadora para enviar, módulo GSM sem um sinal adequado para o envio ou número sem crédito disponível para o envio da

mensagem, pode-se utilizar o *LED* como uma alternativa para verificar se o número possui ou não crédito para os envios da mensagem.

Após o envio do SMS, o programa ligará para o número cadastrado e realiza as verificações nas respostas do módulo para saber se a ligação foi atendida e desligada. Infelizmente, todas as operadoras no Brasil não enviam mais o código de ligação não atendida, elas encaminham para a caixa postal e enviam o código como se a chamada fosse realizada. Portanto, isso poderia gerar uma falha no alerta pela ligação. Para evitar esse problema, foi adicionada uma espera para verificar a conclusão da ligação. Se a chamada não for atendida em, aproximadamente, vinte segundos contados desde o início da ligação, o programa finaliza a chamada e realiza uma nova ligação até o usuário atendê-la dentro do tempo de espera e finalizá-la.

Os alertas utilizados nesse projeto são através da ligação e do envio da mensagem. A ligação é o principal alerta, visto que as mensagens não são totalmente confiáveis, ocorrendo casos de não ser possível enviar a mensagem, como o número estar sem crédito ou problemas da própria operadora em sua transmissão, o que prejudica a comunicação com o usuário.

A Figura 20 ilustra o alerta de uma mensagem que não foi possível ser enviada por problemas com a operadora.

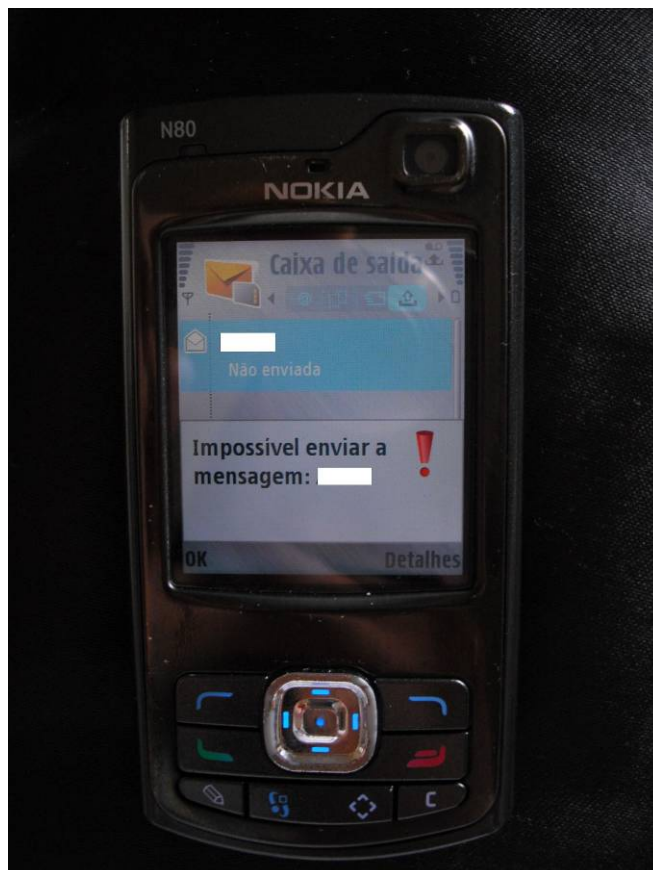


Figura 20 – Impossível enviar mensagem
(Fonte: o autor)

Após a ligação concluída com sucesso, o módulo espera outro sensor ser ativado ou a ligação do usuário para desativar o monitoramento dos sensores. Quando desativado e reativado o monitoramento, os sensores que foram verificados anteriormente são zerados para haver sua verificação.

Para maiores detalhes sobre o código fonte do software, o mesmo se encontra no Apêndice A dessa monografia, já com os comentários para seu fácil entendimento.

4 Resultados obtidos

Foram testadas diversas funcionalidades do software, cada componente e sua integração com os demais, além das tensões nos pontos de saída para outros componentes.

A Figura 21 demonstra a tensão medida diretamente na fonte que simula a bateria do carro.



Figura 21 – Tensão da fonte de 12VDC
(Fonte: o autor)

A Figura 22 mostra a tensão de saída do primeiro regulador de tensão, o LM350.

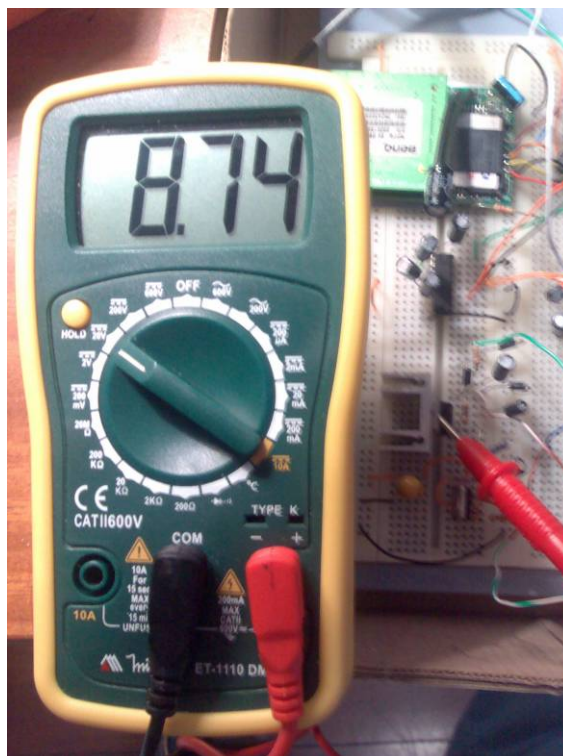


Figura 22 – Tensão de saída medida no LM350

(Fonte: o autor)

A tensão de saída no LM350 é de aproximadamente 8,7 V. Porém, ao passar pela sequência de diodos, sua tensão cai para 8,2 V, visto que os diodos geram uma queda de tensão entre 0,3 a 0,6 V, sendo esse valor ideal para carregar as pilhas recarregáveis. Além disso, é esse regulador de tensão que alimenta os outros dois reguladores que serão detalhados a seguir.

A Figura 23 ilustra a tensão de saída no regulador de tensão LM7805.

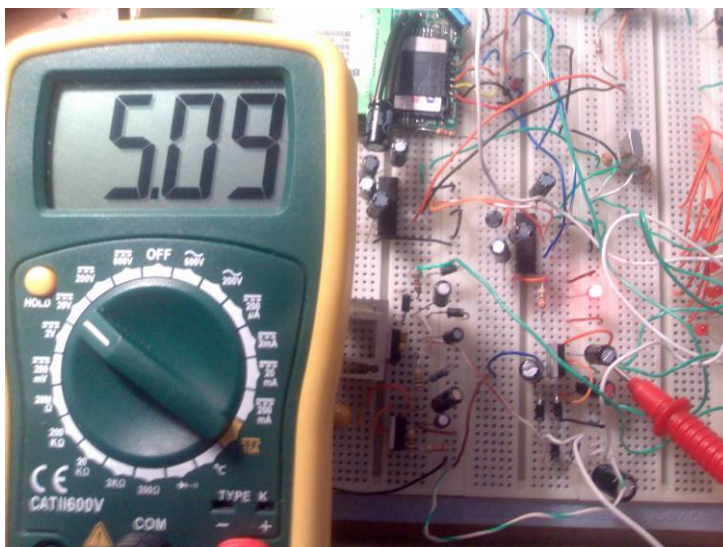


Figura 23 – Tensão de saída no LM7805

(Fonte: o autor)

Essa é a tensão que alimenta o microcontrolador. Quando o microcontrolador é alimentado ela cai para, aproximadamente, 5 V devido a corrente drenada pelos seus componentes.

A Figura 24 demonstra a tensão de saída no regulador LM317T, adaptado para fornecer voltagem necessária ao módulo GSM.

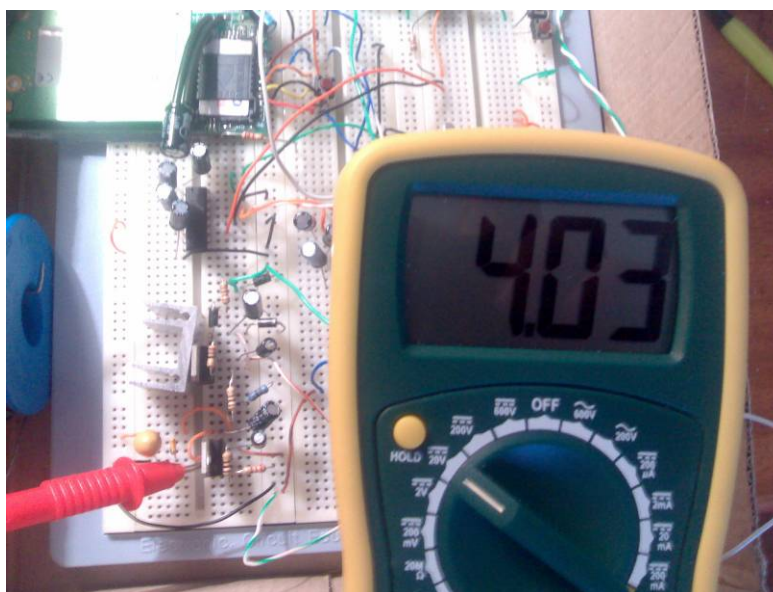


Figura 24 – Tensão de saída do LM317T

(Fonte: o autor)

Foram realizados testes também na tensão da bateria recarregável, que na verdade é um conjunto de seis pilhas recarregáveis em série, para verificar se ela estava carregando corretamente.

Sabe-se que seis pilhas AA em série possuem 1,2V cada pilha, totalizando 7,2 V de tensão média, sendo que quando estão totalmente carregadas, cada pilha pode chegar até 1,4 V em média, totalizando 8,4 V de tensão.

Na Figura 25 podemos ver a voltagem da bateria quase descarregada.



Figura 25 – Tensão inicial da bateria recarregável
(Fonte: o autor)

A bateria ficou ligada por, aproximadamente dez horas e apresentou a seguinte voltagem, ilustrada na Figura 26.

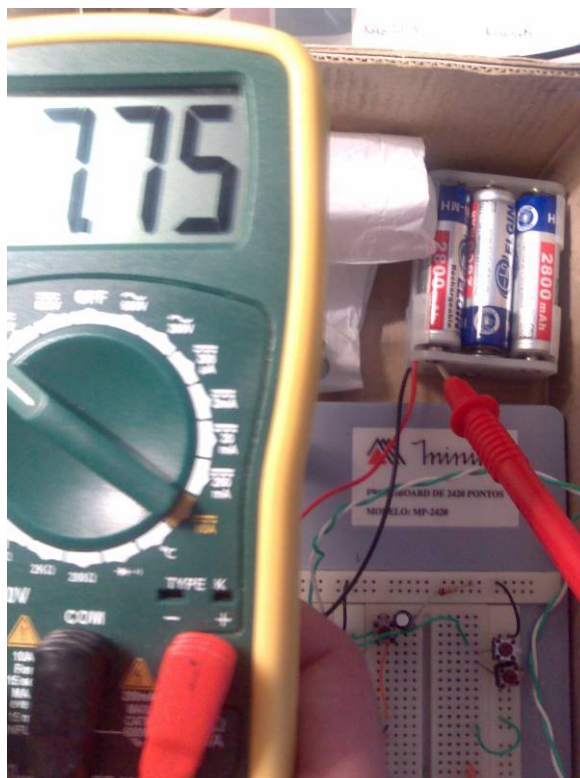


Figura 26 – Tensão da bateria recarregada por 8 horas aproximadamente
(Fonte: o autor)

Portanto, os testes realizados demonstram sucesso no carregamento da bateria recarregável, sendo ele lento, o que otimiza a vida útil das pilhas e ainda não sobrecarrega a corrente do circuito.

Contudo, o software apresentou alguns problemas na comunicação entre o módulo GSM e o microcontrolador. Foi verificado que a função `gets()` armazenava valores que não eram referentes aos comandos enviados. Para corrigir esse problema, foi criada a função `reset_rx()` que altera o ponteiro dos caracteres que ainda não foram lidos para zero, fazendo com que o microcontrolador entenda que não há caractere não lido, gravando por cima dos anteriores. Essa função é utilizada sempre antes de um `printf()` ou `puts()`, que enviam os comandos ao módulo GSM. Após esse ajuste, o software apresentou uma taxa de erro na verificação dos comandos muito menor do que

anteriormente. Pode-se verificar essa alteração no Anexo I, com maiores detalhes do código fonte.

5 Conclusões

Este projeto tem como finalidade realizar um protótipo que alerte o usuário caso algum dos sensores simulados forem acionados, através de um módulo GSM, informando-o pelo seu próprio celular. Ele não foi projetado para realizar alertas sonoros ou visuais, pois sua intenção é de ser silencioso. Porém, pelo microcontrolador possuir diversos pinos de entrada e saída, o projeto pode ser adaptado para incluir outros circuitos que realizem esses tipos de avisos, ou pode ser utilizado em conjunto com outros tipos de alarmes veiculares que realizam essas funções.

O preço final do protótipo é de, aproximadamente, R\$290,00. Esse valor é acessível para uma boa parte da população, sendo que vários componentes utilizados são necessários apenas ao protótipo para verificação e demonstração, como a fonte chaveada, que será substituída pela bateria do carro, vários *LEDs*, entre outros. Deve ser considerado também que os valores de diversos componentes foram retirados do mercado de Brasília, em pequenas quantidades e sendo o comprador uma pessoa física. Esses valores podem se tornar bem mais viáveis caso sejam comprados vários componentes de lojas especializadas, principalmente sendo pessoa jurídica e para o módulo GSM, que é o componente que mais encarece o produto final, visto que o seu valor é quase 50% do valor do projeto e sendo adquiridos vários diretamente com uma fábrica ou revendedora seu custo é reduzido quase pela metade, tornando esse projeto bem mais acessível.

Para maiores detalhes sobre os valores de cada componente utilizado e seu valor total, consulte a Tabela B, que se encontra no Apêndice B desta monografia.

Conclui-se que o objetivo de construir um protótipo para realizar a comunicação entre o veículo e seu usuário através de um módulo GSM, simulando os sensores do veículo, foi realizado com sucesso, sendo viável a um relativo baixo custo, adaptável, por ser modularizado e possuir diversos pinos de entrada e saída e possui ainda diversas possibilidades de evolução e integração com demais dispositivos para melhorar ainda mais essa comunicação.

Durante o desenvolvimento do protótipo foram encontradas várias dificuldades, muitas delas sanadas ao decorrer do projeto e outras são consideradas propostas futuras para sua maior efetividade e funcionalidade, sendo detalhadas a seguir.

5.1 *Dificuldades Encontradas*

Neste capítulo são descritas as dificuldades encontradas, explicando a causa da troca de componentes e suas soluções, possíveis melhorias no projeto como sugestões para propostas futuras do mesmo e melhorias do produto.

5.1.1 Celular

No protótipo inicial a comunicação entre o circuito e o celular do usuário seria realizada utilizando um celular com suporte a comandos AT conectado no próprio circuito. Atualmente, existem poucos celulares que suportam comandos AT e possuam um cabo de dados serial, pois os novos

celulares utilizam para conexão o cabo USB. Além disso, os celulares pesquisados que possuem cabo de dados serial e suporte a comandos AT não são mais produzidos pelas suas fabricantes, como os modelos s25, s35i, s45 e o c35 da Siemens. Assim sendo, para utilizar um celular com o circuito seria necessário encontrar um que possua o suporte aos comandos AT necessários e que utilize um cabo de dados serial ou fazer uma adaptação do cabo de um modelo novo de celular, o que se tornaria inviável, pois ficaria dependente de um modelo específico que, futuramente, também será descontinuado, devendo fazer outra modificação e ajustes no circuito, tornando-o ineficiente.

A Figura 27 ilustra o s40, um dos modelos de celular descontinuados pela Siemens que suporta alguns comandos AT, e os demais componentes contidos em sua embalagem que o acompanhavam de fábrica, assim como o cabo de dados serial com o conector DB-9.



Figura 27 – Celular Siemens modelo S40 e acessórios.

(Fonte:

<http://www.hardwarezone.com/img/data/articles/2001/327/cameinthebox.jpg>

Acessado em 10/09/09)

Outro problema encontrado com a utilização de um celular para a comunicação é que o celular tem funções limitadas de comandos AT, geralmente possuindo apenas comandos básicos, variando de acordo com os modelos e fabricantes, restringindo a funcionalidade do projeto, capacidade de evolução e utilização de novas formas de comunicação com o usuário.

Conclui-se que a utilização de um celular poderia comprometer o andamento do projeto e restringiria suas funções. Para solucionar esse problema, foi pesquisado um novo dispositivo que realizaria essa comunicação sem restrições de hardware e de comandos e foi encontrado o Módulo Wireless GSM.

5.1.2 MC35i

O primeiro módulo GSM pesquisado foi o MC35i Siemens Celular Engine da Siemens. Na sua aquisição foi verificado a existência de um terminal, o TC35i, da própria Siemens, que contém todos os componentes necessários para o funcionamento do módulo, mas foi informado que sua utilização não era obrigatória, podendo utilizar o módulo MC35i com algumas adaptações em sua conexão.

Além disso, como a porta serial do terminal era adaptada para a comunicação com um computador através da porta serial RS-232, utilizando o conector DB-9, os níveis de tensão para essa comunicação são diferentes dos níveis utilizados pelo microcontrolador. Assim sendo, seria necessário adaptar a conexão do terminal com os níveis de tensão utilizados pelo microcontrolador, sendo a sua interface fechada, o que dificulta ainda mais essa adaptação. Além disso, muitos dos componentes do terminal não seriam utilizados nesse projeto e

seu custo é elevado, podendo inviabilizar o propósito do projeto. Portanto, foi decidido que seria adquirido apenas o módulo GSM.

Após a sua aquisição, foi verificado que o conector utilizado pelo módulo é do tipo ZIF¹⁰ FH12-40S 0.5 SH da HIROSE. Pela especificação, pode-se verificar que o conector possui 40 pinos e o espaçamento entre eles é de apenas 0,5mm. Por ser um conector específico, sua utilização é limitada e sua disponibilidade baixa.

Para tentar realizar a integração, foi adquirido o conector específico e foi analisada a possibilidade de se realizar um circuito impresso com o espaçamento desse conector para ser encaixado no mesmo, expandindo-os para outras vias do circuito impresso, utilizando-as para conectar ao *protoboard*, porém foi descartada, pelo prazo de criação da placa, preço e problemas obtidos com uma placa produzida que não estava com o espaçamento correto.

A Figura 28 ilustra o conector utilizado para fazer essa adaptação.



Figura 28 – Conector para adaptação do módulo.

(Fonte: o autor)

¹⁰ ZIF, um padrão utilizado no design de soquetes de circuitos integrados, criado para evitar problemas causados pela força aplicada na inserção e extração de CIs. Adaptado de <<http://pt.wikipedia.org/wiki/ZIF>>. Acessado em 09/09/2009.

Posteriormente, houve a tentativa de uma adaptação utilizando fios rígidos a serem conectados com o circuito, no *protoboard*. Essa adaptação não foi possível por causa do espaçamento entre os pinos, mesmo utilizando equipamentos específicos, como a lupa eletrônica e um ferro de solda de bico fino para soldar esses componentes, além do auxílio de uma pessoa mais experiente com soldagem. Nem mesmo empresas de equipamentos eletrônicos aceitaram o serviço.

Por todos esses empecilhos, o módulo MC35i foi descartado do projeto, adquirindo outro módulo com conectores próprios para a integração com placas e *protoboards* através da soldagem de fios.

5.1.3 Níveis de tensão entre o microcontrolador e módulo GSM

Outra dificuldade analisada foi na diferença entre os níveis de tensão dos pinos do módulo GSM e do microcontrolador, pois um funciona em nível 3.3-V LVTTTL e o outro em TTL, respectivamente. Primeiramente, foi realizada a tentativa de adaptar a voltagem através de resistores como divisores de tensão. Porém, a impedância dos mesmos não ficou igual, havendo problemas na comunicação e uma alta taxa de erro. O mesmo ocorreu ao utilizar um regulador de tensão para adaptar o nível de tensão do microcontrolador para o módulo GSM.

Assim, foi verificada, utilizando o MAX232, a comunicação do microcontrolador e do módulo GSM com o computador, para ver as respostas de cada um e taxas necessárias para o sincronismo. Esse teste foi realizado com sucesso e, como já era utilizado um MAX232 para cada componente se

comunicar com o computador, foi adaptado a conexão entre ambos, fazendo com que um se comunique com o outro, sendo resolvido esse problema com sucesso.

5.1.4 Comunicação com o Módulo GSM

A comunicação com o módulo GSM BenQ M22 não foi facilmente realizada, pois ele envia em todas as suas respostas os caracteres de fim de linha e retorno duas vezes, uma antes do texto de resposta e outra depois do texto. Assim, foi necessário adaptar o programa para eliminar os caracteres iniciais para conseguir analisar o texto de resposta para a sua verificação. Além disso, como a comunicação entre os componentes é assíncrona, foi necessário adaptar o programa para esperar a resposta do módulo e verificar se o módulo está enviando algum dado manualmente. Foram realizados diversos testes e verificações para corrigi-los para minimizar e corrigir os erros ocorridos.

5.2 Propostas Futuras

Existem várias aplicações, melhorias e complementos para o projeto. Algumas sugestões para projetos futuros e melhoria do mesmo são descritas a seguir.

5.2.1 Interação com componentes do veículo

A interação com componentes do veículo tornaria o circuito mais semelhante aos alarmes convencionais e dispensaria a sua utilização em conjunto com alarmes convencionais, tornando-o um módulo completo. Essa interação poderia utilizar os circuitos elétricos, caso existam.

Alguns exemplos de interações são: o acionamento do vidro elétrico e das travas da porta para fechar os vidros e trancar as portas ao alarme ser acionado e destrancá-las ao desativar o alarme, a utilização dos circuito de setas para piscar uma vez quando for acionado e várias vezes quando disparado e o desligamento do som automotivo quando acionado.

5.2.2 Integração com software no Celular (Java)

A constante evolução dos celulares possibilitou que a maioria deles integrasse aplicativos diversos, muitos em linguagem Java. Assim sendo, para uma maior comodidade e melhor utilização do projeto, poderia ser realizada uma integração entre o software no celular do usuário e o projeto, contendo funções para ligar ou desligar o alarme, ativar sensores, ativar ou desativar sirene, entre outros. Além disso, o usuário pode ser alertado sobre qualquer alteração de estado do alarme através do próprio programa sendo executado no celular.

Deverá ser analisado se é obrigatória a utilização de um servidor para essa aplicação ou apenas a utilização da transmissão dos pacotes via GPRS, uma opção que o presente módulo GSM suporta.

5.2.3 Interação com módulo GPS

A utilização de um módulo GPS poderia ampliar a utilização do circuito para monitorar a localização do veículo, podendo ser utilizado quando o usuário esqueceu aonde estacionou o veículo, se o seu veículo foi furtado, podendo até mesmo ter sido guinchado ou para o acionamento do circuito, caso haja um deslocamento do veículo para avisar antes mesmo que isso aconteça. Essa solução pode ser utilizada de várias formas, como enviando apenas as

coordenadas através de mensagem e o usuário teria que utilizar algum aplicativo, como o Google maps, para saber onde o veículo está ou, até mesmo, fazer um aplicativo com o próprio celular, para indicar o local, o que provavelmente utilizaria a proposta anterior, que integra um software para o celular.

5.2.4 Interação com sensores específicos

Como esse projeto apenas simula os sensores, é recomendável a utilização e sua adaptação com sensores reais, podendo utilizar novos tipos, de acordo com seu objetivo.

Referências Bibliográficas

NICOLOSI, Denys E. C. *Microcontrolador 8051 Detalhado*. São Paulo: Ed. Érica, 2004

STEINER, Crag. *The 8051/8052 Microcontroller: Architecture, Assembly Language, and Hardware Interfacing*. Ed. Universal Pub, 2005.

THOMAZINI, Daniel; ALBUQUERQUE, Pedro Urbano Braga de. *Sensores industriais: fundamentos e aplicações*. São Paulo: Érica, 2005.

ZELENOVSKY, Ricardo; MENDONÇA, Alexandre. *Microcontroladores: Programação e Projeto com a Família 8051*. São Paulo: Mz editora, 2005.

CAVALCANTI, Diogo Cogo. Trabalho de conclusão do Curso de Engenharia da Computação do Centro Universitário de Brasília. *Alarme Automotivo Integrado ao Aparelho Celular*. Brasília, 2004. Orientador Prof. M.C. Maria Marony Sousa Farias Nascimento.

NASCIMENTO, Ivan Sampaio. Trabalho de conclusão do Curso de Engenharia da Computação do Centro Universitário de Brasília. *Sistema de alarme automotivo que integra transdutor acústico/elétrico e celular*. Brasília, 2007. Orientador Prof. Aderlon Marcelino Queiroz.

EPOPEIA Eletrônica. *Epopéia Eletrônica – Kit de Estudo 8051*. Disponível em: <http://www.epopeiaeletronica.com/index.php?option=com_content&task=view&id=9&Itemid=15>. Acesso em: 14 jun. 2008.

Escola Politécnica da USP – COMUNICAÇÃO SERIAL. Disponível em: <http://www.poli.usp.br/d/pcs2529/index_arquivos/2529e062005.pdf>. Acesso em: 04 jun. 2009.

GPS. Disponível em: <<http://www.tbat.com.br/GPS.htm>>. Acesso em: 08 mai. 2009.

HOWSTUFFWORKS – Como funcionam os alarmes de carros, Disponível em <<http://carros.hsw.uol.com.br/alarmes-dos-carros.htm>>. Acesso em 17 de ago. 2009.

MOHR, Hari. *O microcontrolador 8051*. Apostila, Depto. de Engenharia Elétrica, UFSC, Florianópolis, SC, 2001. Disponível em: <www.das.ufsc.br/~werner/eel7030/8051/Apostila8051Hari.pdf>. Acesso em: 20 out. 2008.

National Marine Electronic Association – NMEA. Disponível em <<http://www.nmea.org>>. Acesso em 03 jul. 2009.

PC MAGAZINE - *AT command set definition*. Disponível em: <http://www.pcmag.com/encyclopedia_term/0,2542,t=AT+command+set&i=38080,00.asp>. Acesso em: 20 out. 2008.

Ricardo Zelenovsky - *Página de Ricardo Zelenovsky / Livro de 8051*. Disponível em: <http://www.ene.unb.br/%7Ezele/Livro_8051/Livro_8051_1.html>. Acesso em: 24 jul. 2008.

SIEMENS Brasil - *Produtos*. Disponível em: <http://www.siemens.com.br/templates/produto.aspx?channel=6578&channel_prioridade=6650&produto=16244>. Acesso em: 05 ago. 2008.

Apêndices

A seguir, encontra-se a versão atual do código fonte do programa utilizado para fazer a interação com o módulo GSM, verificação dos sensores e demais funcionalidades relacionadas ao projeto.

APÊNDICE I – Código fonte do programa

/*-----

Programa.c: Programa que realiza a configuração do módulo GSM, estabelece uma comunicação com o microcontrolador e gerencia todo o funcionamento do circuito.

Desenvolvido por: Cristiano Rabelo Isaac

Email: cristianoisaac@gmail.com

Para utilização, favor consultar o desenvolvedor.

-----*/

```
#include <at89x52.H>
```

```
#include <stdio.h>
```

```
#include "sio_Final.H"
```

```
#include <string.h>
```

```
void delay_ms(unsigned int ms);
```

```
void padrao_modulo();
```

```
void ligar ();
```

```
void mensagem(unsigned int sens);
```

```
char _getkey (void);
```

```
char putchar (char c);
```

```
void limpastring(void);
```

```

unsigned short int i,t, PortaAntes=0xFF00, PortaAtual, BitAlterado;

unsigned char retorno[40],c; //unificado como variável global de retorno para
economizar no espaço de memória reservado a variáveis

```

```

void main (){

unsigned short int Verificados=0xFF00,Verificar,BitAtual,Aciona=0;

    //P2_4=0; //Indica que o microcontrolador está operacional.

    delay_ms(8000); //Espera inicialização módulo.

    com_initialize (); //inicializa a interrupção da porta serial. /* initialize
interrupt driven serial I/O */

    com_baudrate (9600); //CONfigura o baudrate para 9600.

    EA = 1; //Habilita as interrupções

    ET0 = 1; //Habilita as interrupções do timer 0.

    TR0 = 1; //Habilita o timer 0.

    padrao_modulo();

    while (1)
    {

        if (P2_5 != 1 ) // Faz a verificação na interrupção da porta serial se
existe algum caractere informado pelo módulo que não foi lido.

        {

            P2_1=0; //Aguardando Resposta do Módulo

            limpastring();

            do

            {

```

```

while(P2_5!=0){};
c=getchar();
P2_5=1;
if((c != '\0' && c != '\r' && c != '\n') )
{
    retorno[t]=c;
    t++;
}
}while(c != '\r' && c != '\n' && t < 38 || t < 1);
P2_1=1; //Resposta Recebida
if (strstr(retorno,"Dono")!=NULL ||
strstr(retorno,"84248528")!=NULL) //Verificar se o número do usuário está ligando
{
    P2_1=1; //Resposta Recebida
    Aciona=1;
    Verificados=0xFF00;
    PortaAntes=0xFF00;
    P2_0=0;
    delay_ms(3000);
    reset_rx();
}
}
while (Aciona == 1)
{
    Verificar = Verificados^(PortaAntes|Verificados);
    if (Verificar != 0)

```

```

{
P2_4=0;//Sensor Habilitado e realizando operações de
resposta

    for (i = 0 ; i < 8 ; i++)
    {
        BitAtual = 1 << i;    /* Pega o Bit de ordem i */
        if ((int)(Verificar&BitAtual) != 0)    /* este bit está
ligado???? */
        {
            mensagem(i);
            delay_ms(1000);
            ligar();
            delay_ms(5000);
        }
    }
Verificados = (Verificados|Verificar); // armazena os bits já
verificados

}
P2_4=1;//Sensor Desabilitado
reset_rx();
if (P2_5==0) // Faz a verificação na interrupção da porta serial
se existe algum caractere informado pelo módulo que não foi lido.
{

    P2_1=0; //Aguardando Resposta do Módulo
    limpastring();
    do

```



```

TR0=0;      //Para a execução do Timer 0

TMOD = ((TMOD & 0xF0)) | 0x01;

PortaAtual=(~P0);

if (((PortaAtual^PortaAntes)&PortaAtual)>=1)

//verifica o estado da porta atual comparando com seu estado anterior

{

    PortaAntes=(PortaAntes|PortaAtual);

    //Atribui o estado anterior ao atual para não verificar os pinos
novamente

}

    TR0 = 1;

}

```

```

void delay_ms(unsigned int ms) //Função de geração de tempo - base 1us

{

    TL2 = 0x66;                //Lower Byte

    TH2 = 0xFC;                //Upper Byte

    RCAP2H = 0xFC;             //Upper byte set value when
reload occurs

    RCAP2L = 0x66;             //Lower Byte set value when
reload occurs

    TR2=1; //Habilita o timer 2.

    while(ms)

    {

        if (TF2==1) //Estouro do timer

        {

```

```

        ms--; // Decrementa a variável

        TF2 = 0;

    }

}

TR2=0; //Desabilita o timer 2.

}

void padrao_modulo(){

    delay_ms(1000);
    P2_1=0; //Aguardando Resposta do Módulo
    printf("AT\r");
    delay_ms(100);
    reset_rx(); // Reseta todos os caracteres recebidos até então.
    printf("ATE0\r");
    delay_ms(100);
    reset_rx(); // Reseta todos os caracteres recebidos até então.
    printf("AT&D0\r");
    delay_ms(100);
    reset_rx(); // Reseta todos os caracteres recebidos até então.
    delay_ms(100);
    printf("AT&C0\r");
    delay_ms(100);
    reset_rx(); // Reseta todos os caracteres recebidos até então.
    delay_ms(100);
    printf("ATV0\r");

```

```

delay_ms(100);

reset_rx(); // Reseta todos os caracteres recebidos até então.

printf("AT$NOSLEEP=1\r");

delay_ms(100);

reset_rx(); // Reseta todos os caracteres recebidos até então.

printf("AT$TRING=5000\r");

delay_ms(100);

reset_rx(); // Reseta todos os caracteres recebidos até então.

delay_ms(100);

printf("ATS4=124\r"); //Seta os caracteres de <LF> como '|'

delay_ms(100);

reset_rx(); // Reseta todos os caracteres recebidos até então.

delay_ms(100);

printf("AT+IPR=9600\r");

delay_ms(100);

reset_rx(); // Reseta todos os caracteres recebidos até então.

delay_ms(100);

printf("AT+CFUN=1\r");

delay_ms(100);

reset_rx(); // Reseta todos os caracteres recebidos até então.

delay_ms(100);

printf("ATS0=1\r"); //Atende automaticamente uma ligação em 3
chamadas.(PÁGINA 158 MÓDULO)

delay_ms(100);

reset_rx(); // Reseta todos os caracteres recebidos até então.

delay_ms(100);

```

```

printf("AT\r");
limpastring();
do
{
    while(P2_5!=0){};
    c=getchar();
    P2_5=1;
    if((c != '\0' && c != '\r' && c != '\n') )
    {
        retorno[t]=c;
        t++;
    }
}while(c != '\r' && c != '\n' && t < 38 || t < 1);
P2_1=1; //Resposta Recebida
while ((strstr(retorno,"0")==NULL) && (strstr(retorno,"0")==NULL))
{
    P2_2=0; //Esperando a resposta como OK
    reset_rx(); // Reseta todos os caracteres recebidos até então.
    printf("AT\r");
    P2_1=0; //Aguardando Resposta do Módulo
    limpastring();
    do
    {
        while(P2_5!=0){};
        c=getchar();

```

```

        P2_5=1;

        if((c != '\0' && c != '\r' && c != '\n') )
        {
            retorno[t]=c;

            t++;
        }

    }while(c != '\r' && c != '\n' && t < 38 || t < 1);

    P2_1=1; //Resposta Recebida
}

P2_2=1; //Resposta OK recebida
delay_ms(2000);

reset_rx();

delay_ms(100);

printf("AT+CLIP=1\r"); //Habilita a Identificação da chamada. VERIFICAR
MELHOR USO.(PÁGINA 233 MÓDULO)

P2_1=0; //Aguardando Resposta do Módulo

limpastring();

do
{
    while(P2_5!=0){};

    c=getchar();

    P2_5=1;

    if((c != '\0' && c != '\r' && c != '\n') )
    {

        retorno[t]=c;

```

```

        t++;
    }
}while(c != '\r' && c != '\n' && t < 38 || t < 1);

P2_1=1; //Resposta Recebida
while (strstr(retorno,"0")==NULL)
{
    reset_rx();
    printf("AT+CLIP=1\r");
    P2_1=0; //Aguardando Resposta do Módulo
    limpastring();
    do
    {
        while(P2_5!=0){};
        c=getchar();
        P2_5=1;
        if((c != '\0' && c != '\r' && c != '\n') )
        {
            retorno[t]=c;
            t++;
        }
    }while(c != '\r' && c != '\n' && t < 38 || t < 1);

    P2_1=1; //Resposta Recebida
}

P2_2=1; //OK recebido
delay_ms(2000);

```

```

    reset_rx();

    printf("AT+CMGF=1\r"); //Formato da mensagem de texto SMS para texto
em vez de PDU (PÁGINA 281 MÓDULO)

    P2_1=0; //Aguardando Resposta do Módulo

    limpastring();

    do

    {

        while(P2_5!=0){};

        c=getchar();

        P2_5=1;

        if((c != '\0' && c != '\r' && c != '\n') )

        {

            retorno[t]=c;

            t++;

        }

    }while(c != '\r' && c != '\n' && t < 38 || t < 1);

    P2_1=1; //Resposta Recebida

    while (strstr(retorno,"0")==NULL)

    {

        reset_rx();

        printf("AT+CMGF=1\r"); //Formato da mensagem de texto SMS para
texto em vez de PDU (PÁGINA 281 MÓDULO)

        P2_1=0; //Aguardando Resposta do Módulo

        limpastring();

        do

        {

            while(P2_5!=0){};

```

```

        c=getchar();

        P2_5=1;

        if((c != '\0' && c != '\r' && c != '\n') )
        {
                retorno[t]=c;

                t++;

        }

    }while(c != '\r' && c != '\n' && t < 38 || t < 1);

    P2_1=1; //Resposta Recebida

}

P2_2=1; //OK recebido

delay_ms(2000);


reset_rx();

printf("AT+CPBW=1,\"84248528\",129,\"Dono\r");//Escreve na memória
padrão do módulo (SIM) (PÁGINA 349)

delay_ms(100);

reset_rx();

delay_ms(100);

limpastring();

}

void ligar ()
{

    reset_rx();

    printf("AT\r");

```



```

delay_ms(1000);

reset_rx();

printf("AT+CHUP\r"); //Desliga qualquer ligação que estiver sendo
realizada.

limpastring();

delay_ms(1000);

reset_rx();

delay_ms(500);

printf("ATD909084248528;\r"); //Realiza a discagem a cobrar (local) para o
número informado.

delay_ms(2000);

reset_rx();

delay_ms(16000);

reset_rx();

printf("AT+CEER\r"); //Verifica se o status da última ligação realizada.

P2_1=0; //Aguardando Resposta do Módulo

limpastring();

do
{
    while(P2_5!=0){};

    c=getchar();

    P2_5=1;

    if((c != '\0' && c != '\r' && c != '\n') )
    {
        retorno[t]=c;

        t++;
    }
}

```

```

}while(c != '\r' && c != '\n' && t < 38 || t < 1);

P2_1=1; //Resposta Recebida

while ((strstr(retorno,"clearing") == NULL) &&
(strstr(retorno,"CLEARING")== NULL))
{
    P2_2=0; //Esperando a resposta como OK

    printf("AT+CHUP\r");

    limpastring();

    delay_ms(1000);

    reset_rx();

    delay_ms(500);

    printf("ATD909084248528;\r");

    delay_ms(2000);

    reset_rx();

    delay_ms(16000);

    reset_rx();

    printf("AT+CEER\r");

    P2_1=0; //Aguardando Resposta do Módulo

    limpastring();

    do
    {
        while(P2_5!=0){};

        c=getchar();

        P2_5=1;

        if((c != '\0' && c != '\r' && c != '\n') )
        {

```

```

        retorno[t]=c;

        t++;

    }

    }while(c != '\r' && c != '\n' && t < 38 || t < 1);

    P2_1=1; //Resposta Recebida

}

P2_2=1; // Clearing Recebido, indicando que a ligação foi recebida e
desligada com sucesso.

    delay_ms(1000);

    reset_rx();

    delay_ms(500);

}

```

```

void mensagem(unsigned int sens){

    //unsigned int j=0;

//    //unsigned char idata msg_ret[50],
    unsigned char idata msg[30];

    switch (sens)

    {

        case 0:

            strcpy(msg,"da Porta esquerda");

            break;

        case 1:

            strcpy(msg,"da Porta direita");

            break;

```

```

    case 2:
        strcpy(msg,"da Porta direita traseira");
        break;
    case 3:
        strcpy(msg,"da Porta esquerda traseira");
        break;
    case 4:
        strcpy(msg,"do capo");
        break;
    case 5:
        strcpy(msg,"ultrasom");
        break;
    case 6:
        strcpy(msg,"da Bateria");
        break;
    case 7:
        strcpy(msg,"nao configurado");
        break;
}

reset_rx();

printf("AT+CMGS=\"06184248528\"\r\n");// (PÁGINA 286MÓDULO)
ATD>SM15; Liga para o 15º número do SIM.

    delay_ms(2000)/*tempo de espera para aparecer o > para enviar a
mensagem*/;

    P2_1=1; //Resposta Recebida

    reset_rx();

```

printf("ALARME ACIONADO - Sensor %s disparado!", msg); //mensagem a ser enviada

delay_ms(2000);

putchar(0x1A);

delay_ms(500);

P2_1=0; //Aguardando Resposta do Módulo

limpastring();

do

{

while(P2_5!=0){};

c=getchar();

P2_5=1;

if((c != '\0' && c != '\r' && c != '\n'))

{

retorno[t]=c;

t++;

}

}while(c != '\r' && c != '\n' && t < 38 || t < 1);

P2_1=1; //Resposta Recebida

if (strstr(retorno,"CMGS")==NULL)

{

P2_3=0;

reset_rx();

delay_ms(100);

}

else

```
{  
    P2_3=1;  
    reset_rx();  
    delay_ms(100);  
}  
}
```

```
char _getkey (void)  
{  
    int k;  
  
    do  
    {  
        k = com_getchar ();  
    }  
    while (k == -1);  
  
    return ((unsigned char) k);  
}
```

```
char putchar (char c)  
{  
    while (com_putchar (c) != 0);  
  
    return (c);  
}
```

```
void limpastring(void)
{
    unsigned short int h;
    unsigned short int fim=t;
    for(h=0;h<fim;h++)
    {
        retorno[h]='\0';
    }
    t=0;
}
```

APÊNDICE II – Preços dos componentes utilizados

A Tabela II relata os componentes utilizados com suas referências, descrições, quantidades e o preço médio no mercado para pessoas físicas.

Tabela 6 – Componentes utilizados, referência e valores. (continuação).

(Fonte: o autor)

Componente	Referência ao circuito	Descrição	Qtd	Preço
6 Pilhas NI-MH 2800 mAH	BT1	7.2 V	1	R\$ 57,00
Capacitor Eletrolítico	C1	4700 uF/25 V	1	R\$ 5,10
Capacitor Eletrolítico	C2, C3, C4	100 uF/25 V	2	R\$ 1,20
Capacitor Eletrolítico	C5, C7	0.1 uF/50 V	2	R\$ 0,60
Capacitor Eletrolítico	C6, C8, C12, C13, C14, C15, C16, C17, C18, C19, C20	1 uF/50 V	11	R\$ 2,20
Capacitor Cerâmico	C9, C10	33 pF/400 V	2	R\$ 1,20
Capacitor Eletrolítico	C11	10 uF/25 V	1	R\$ 0,25
Capacitor de Tântalo	C21	100 uF/35 V	1	R\$ 1,10
Capacitor Cerâmico	C22	0.22 uF/30 V	1	R\$ 0,30
Diodo Retificador	D1, D2, D3	1N4007	3	R\$ 1,20
LED Vermelho	D4, D5, D6, D7, D8, D9, D10, D11, D12	LED	9	R\$ 1,80
Phone Jack para fonte	J1	PHONE JACK	1	R\$ 1,60
Resistor Cerâmico	R1, R6	220 / 1/4 W	2	R\$ 0,20
Resistor Cerâmico	R2	12 / 1/4 W	1	R\$ 0,10
Resistor Cerâmico	R3	18 / 1/4 W	1	R\$ 0,10
Resistor Cerâmico	R4	10 / 1/4 W	1	R\$ 0,10
Resistor Cerâmico	R5,R7,R8	100 / 1/4 W	3	R\$ 0,30

Resistor Cerâmico	R9,R19	10K / 1/4 W	2	R\$ 0,20
Resistor Cerâmico	R10, R11, R12, R13, R14, R15, R16, R17	560 / 1/4 W	8	R\$ 0,80
Resistor Cerâmico	R18	22 / 1/4 W	1	R\$ 0,10
Resistor Cerâmico	R20	3.3 / 1/4 W	1	R\$ 0,10
Chave Pushbutton sem trava	SW1, SW2, SW3, SW4, SW5, SW6, SW7, SW8, SW9	PUSHBUTTON	9	R\$ 18,00
Switch	S1	SWITCH	1	R\$ 5,00
Regulador de tensão	U1	LM350	1	R\$ 4,40
Regulador de tensão	U2	LM317T	1	R\$ 1,60
Regulador de tensão	U3	LM7805C	1	R\$ 1,00
Microcontrolador	U4	AT89S52	1	R\$ 13,00
CI MAX232	U5, U6	MAX232	2	R\$ 6,00
Módulo Wireless GSM	U7	BenQ M22	1	R\$ 131,00
Conector para SIMCARD	U8	SIMCARD	1	-
Cristal de Quartzo	Y1	11,0592 MHz	1	R\$ 1,60
Cartão SIM	-	-	1	R\$ 10,00
Dissipador pequeno	-	-	1	R\$ 1,20
Fonte Chaveada	-	12 VDC / 3,2 A	1	R\$ 23,00
Fio fino	-	1 M	1	R\$ 1,00
Total				R\$ 292,35

ANEXOS

A seguir serão inseridos os anexos dos datasheets dos principais componentes utilizados para a construção do protótipo que são necessários para maior detalhamento da sua utilização, além do código fonte adaptado para o projeto para a sincronia da comunicação entre o módulo GSM e o microcontrolador.

ANEXO I – Código fonte para comunicação entre o modulo GSM e microcontrolador

```
/*-----
```

SIO.C: Serial Communication Routines.

Copyright 1995-2002 KEIL Software, Inc.

```
-----*/
```

```
#include <at89x52.h>
```

```
#include <string.h>
```

```
#include "sio.h"
```

```
/*-----
```

Notes:

The length of the receive and transmit buffers must be a power of 2.

Each buffer has a next_in and a next_out index.

If next_in = next_out, the buffer is empty.

$(\text{next_in} - \text{next_out}) \% \text{buffer_size}$ = the number of characters in the buffer.

```
-----*/
```

```

#define TBUF_SIZE 32      /*** Must be one of these powers of 2
(2,4,8,16,32,64,128) ***/

#define RBUF_SIZE 64      /*** Must be one of these powers of 2
(2,4,8,16,32,64,128) ***/


#define TBUF_SPACE idata  /*** Memory space where the transmit buffer
resides ***/

#define RBUF_SPACE idata  /*** Memory space where the receive buffer
resides ***/


#define CTRL_SPACE data    /*** Memory space for the buffer indexes ***/


/*-----
-----*/

#if TBUF_SIZE < 2
#error TBUF_SIZE is too small. It must be larger than 1.
#elif TBUF_SIZE > 128
#error TBUF_SIZE is too large. It must be smaller than 129.
#elif ((TBUF_SIZE & (TBUF_SIZE-1)) != 0)
#error TBUF_SIZE must be a power of 2.
#endif

#if RBUF_SIZE < 2
#error RBUF_SIZE is too small. It must be larger than 1.
#elif RBUF_SIZE > 128
#error RBUF_SIZE is too large. It must be smaller than 129.
#elif ((RBUF_SIZE & (RBUF_SIZE-1)) != 0)

```

```
#error RBUF_SIZE must be a power of 2.
```

```
#endif
```

```
/*-----  
-----*/
```

```
static TBUF_SPACE unsigned char tbuf [TBUF_SIZE];
```

```
static RBUF_SPACE unsigned char rbuf [RBUF_SIZE];
```

```
static CTRL_SPACE unsigned char t_in = 0;
```

```
static CTRL_SPACE unsigned char t_out = 0;
```

```
static CTRL_SPACE unsigned char r_in = 0;
```

```
static CTRL_SPACE unsigned char r_out = 0;
```

```
static bit ti_restart = 0; /* NZ if TI=1 is required */
```

```
/*-----  
-----*/
```

```
static void com_isr (void) interrupt 4
```

```
{
```

```
/*-----
```

```
Received data interrupt.
```

```
-----*/
```

```
if (RI != 0)
```

```
{
```

```
RI = 0;
```

```
if (((r_in - r_out) & ~(RBUF_SIZE-1)) == 0)
```

```
{
```

```
  rbuf [r_in & (RBUF_SIZE-1)] = SBUF;
```

```
  r_in++;
```

```
}
```

```
    P2_5=0;
```

```
}
```

```
/*-----
```

```
Transmitted data interrupt.
```

```
-----*/
```

```
if (TI != 0)
```

```
{
```

```
  TI = 0;
```

```
if (t_in != t_out)
```

```
{
```

```
    SBUF = tbuf [t_out & (TBUF_SIZE-1)];
```

```
    t_out++;
```

```
    ti_restart = 0;
```

```
}
```

```
else
```

```
{
```

```
  ti_restart = 1;
```

```

    }

    P2_5=1;
}

}

/*-----
-----*/

#pragma disable

void com_initialize (void)
{
/*-----

Setup TIMER1 to generate the proper baud rate.
-----*/

com_baudrate (9600);

/*-----

Clear com buffer indexes.
-----*/

t_in = 0;
t_out = 0;

r_in = 0;
r_out = 0;

```

```

/*-----
Setup serial port registers.
-----*/

SM0 = 0; SM1 = 1; /* serial port MODE 1 */
SM2 = 0;
REN = 1;          /* enable serial receiver */

RI = 0;           /* clear receiver interrupt */
TI = 0;           /* clear transmit interrupt */
ti_restart = 1;

ES = 1;           /* enable serial interrupts */
PS = 0;           /* set serial interrupts to low priority */
}

/*-----
-----*/

#pragma disable

void com_baudrate (
    unsigned baudrate)
{
    /*-----
Clear transmit interrupt and buffer.
-----*/

```



```

TI = 0;          /* clear transmit interrupt */
t_in = 0;        /* empty transmit buffer */
t_out = 0;

/*-----
Set timer 1 up as a baud rate generator.
-----*/

TR1 = 0;         /* stop timer 1 */
ET1 = 0;         /* disable timer 1 interrupt */

PCON |= 0x80;    /* 0x80=SMOD: set serial baudrate doubler */

TMOD &= ~0xF0;   /* clear timer 1 mode bits */
TMOD |= 0x20;    /* put timer 1 into MODE 2 */

TH1 = (unsigned char) (256 - (XTAL / (16L * 12L * baudrate)));

TR1 = 1;         /* start timer 1 */
}

/*-----
-----*/

#pragma disable

char com_putchar (
    unsigned char c)

```

```

{
/*-----
If the buffer is full, return an error value.
-----*/

if (com_tbuflen () >= TBUF_SIZE)
    return (-1);

/*-----

Add the data to the transmit buffer. If the
transmit interrupt is disabled, then enable it.
-----*/

tbuf [t_in & (TBUF_SIZE - 1)] = c;
t_in++;

if (ti_restart)
{
    ti_restart = 0;
    TI = 1;          /* generate transmit interrupt */
}

return (0);
}

/*-----
-----*/

#pragma disable

```

```

int com_getchar (void)
{
    if (com_rbuflen () == 0)
        return (-1);

    return (rbuf [(r_out++) & (RBUF_SIZE - 1)]);
}

/*-----
-----*/

#pragma disable

unsigned char com_rbuflen (void)
{
    return (r_in - r_out);
}

/*-----
-----*/

#pragma disable

unsigned char com_tbuflen (void)
{
    return (t_in - t_out);
}

```

```
/*-----  
-----*/
```

void reset_rx (void) //Criado para zerar o conteúdo anterior transmitido pelo módulo.

```
{  
    r_out=r_in=0;  
    t_out=t_in=0;  
    P2_5=1;  
}
```

ANEXO II – Datasheet AT89S52

Esse anexo descreve as principais funcionalidades utilizadas pelo microcontrolador, especificando os níveis de tensão em cada pino e o funcionamento de cada um com suas funções específicas, caso existam.

Esse anexo encontra-se na mídia de CD entregue juntamente com a versão impressa da monografia, dentro da pasta Anexos.

ANEXO III – Datasheet BenQ M22 Rev. 0.2

É detalhado nesse anexo os pinos do módulo GSM BenQ M22 e suas funcionalidades, assim como seus níveis de tensão e demais especificações técnicas.

Esse anexo encontra-se na mídia de CD entregue juntamente com a versão impressa da monografia, dentro da pasta Anexos.

ANEXO IV – BenQ M22 AT Command Rev. 0.3.0

Todos os comandos, respostas do módulo e seu funcionamento podem ser verificados nesse anexo. Ele está armazenado na mídia de CD entregue juntamente com a versão impressa da monografia, dentro da pasta Anexos.

ANEXO V – Datasheet LM317

Esse datasheet especifica alguns modos que esse regulador de tensão pode funcionar, detalhando o seu funcionamento e os circuitos necessários para a sua realização. Ele está armazenado na mídia de CD entregue juntamente com a versão impressa da monografia, dentro da pasta Anexos.

ANEXO VI – Datasheet MAX232

Finalmente, o datasheet do MAX232 descreve os pinos e circuitos necessários para a sua comunicação entre os dispositivos suportados, além de mostrar a tensão suportada em cada pino. Ele se encontra na mídia de CD entregue juntamente com a versão impressa da monografia, dentro da pasta Anexos.